

C FO 13009 VS/ed
09/170 656
XAV 2787

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application:

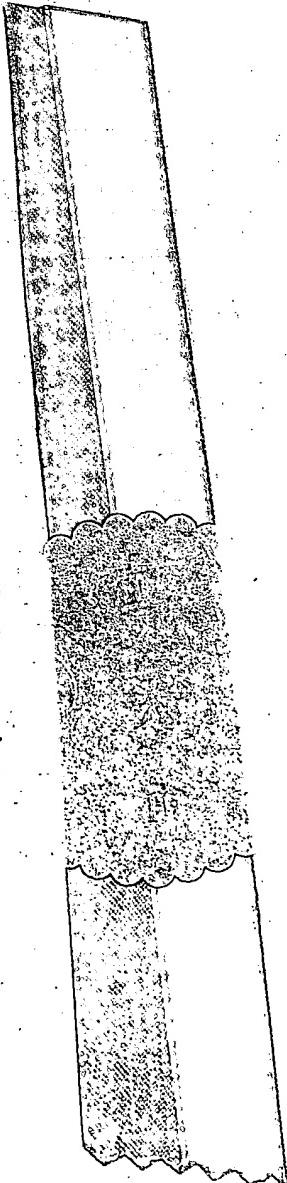
1997年10月14日

出願番号
Application Number:

平成9年特許願第280750号

出願人
Applicant(s):

キヤノン株式会社

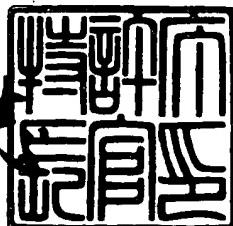


CERTIFIED COPY OF
PRIORITY DOCUMENT

1998年11月6日

特許庁長官
Commissioner,
Patent Office

佐山 建



【書類名】 特許願
【整理番号】 3600034
【提出日】 平成 9年10月14日
【あて先】 特許庁長官 荒井 寿光 殿
【国際特許分類】 G06F 17/60
【発明の名称】 情報処理装置及びその方法、及びそのプログラムを記憶
した記憶媒体
【請求項の数】 17
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会社
内
【氏名】 若井 聖範
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会社
内
【氏名】 ジェヤチャンドラン スレッシュ
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会社
内
【氏名】 上田 直子
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会社
内
【氏名】 須田 アルナ・ローラ
【特許出願人】
【識別番号】 000001007
【郵便番号】 146
【住所又は居所】 東京都大田区下丸子3丁目30番2号
【氏名又は名称】 キヤノン株式会社

【代表者】 御手洗 富士夫

【電話番号】 03-3758-2111

【代理人】

【識別番号】 100069877

【郵便番号】 146

【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会社
内

【弁理士】

【氏名又は名称】 丸島 儀一

【電話番号】 03-3758-2111

【手数料の表示】

【予納台帳番号】 011224

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9703271

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置及びその方法、及びそのプログラムを記憶した記憶媒体

【特許請求の範囲】

【請求項 1】 処理対象情報の属性を指定する属性指定手段と、
処理対象情報に対する処理を指定する処理指定手段と、
実行すべき命令を設定する設定手段と、
前記属性指定手段により指定された属性の処理対象情報に対して前記処理指定手段により指定された処理が実行された場合に、前記設定手段より設定された命令を実行する実行手段とを備えることを特徴とする情報処理装置。

【請求項 2】 有効期限を設定する期限設定手段と、該期限設定手段により設定された期限を過ぎた場合に、前記実行手段による命令の実行を禁止する禁止手段とを備えることを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】 プリンタの状態を指定する状態指定手段と、該状態指定手段により指定された状態の場合に、前記実行手段による命令の実行を許可する許可手段とを備えることを特徴とする請求項 1 記載の情報処理装置。

【請求項 4】 前記処理対象情報の属性は、処理対象情報名、処理対象情報の送信装置名、処理対象情報の送信者名の少なくとも1つを含むことを特徴とする請求項 1 記載の情報処理装置。

【請求項 5】 前記処理は、処理対象情報の受信、送信、印刷の少なくとも1つを含むことを特徴とする請求項 1 記載の情報処理装置。

【請求項 6】 前記命令は、通知または処理対象情報に対する処理を行なうための命令を含むことを特徴とする請求項 1 記載の情報処理装置。

【請求項 7】 前記命令は、処理対象情報に対する印刷または保留の少なくとも1つを行なうための含むことを特徴とする請求項 6 記載の情報処理装置。

【請求項 8】 前記期限設定手段により設定された期限を過ぎた場合に、前記設定手段により設定された命令を削除する管理手段を有することを特徴とする請求項 2 記載の情報処理装置。

【請求項 9】 处理対象情報の属性を指定する属性指定工程と、
処理対象情報に対する処理を指定する処理指定工程と、
実行すべき命令を設定する設定工程と、
前記属性指定工程により指定された属性の処理対象情報に対して前記処理指定
工程により指定された処理が実行された場合に、前記設定工程より設定された命
令を実行する実行工程とを備えることを特徴とする情報処理方法。

【請求項 10】 有効期限を設定する期限設定工程と、該期限設定工程によ
り設定された期限を過ぎた場合に、前記実行工程による命令の実行を禁止する禁
止工程とを備えることを特徴とする請求項 9 記載の情報処理方法。

【請求項 11】 プリンタの状態を指定する状態指定工程と、該状態指定工
程により指定された状態の場合に、前記実行工程による命令の実行を許可する許
可工程とを備えることを特徴とする請求項 9 記載の情報処理方法。

【請求項 12】 前記処理対象情報の属性は、処理対象情報名、処理対象情
報の送信方法名、処理対象情報の送信者名の少なくとも1つを含むことを特徴と
する請求項 9 記載の情報処理方法。

【請求項 13】 前記処理は、処理対象情報の受信、送信、印刷の少なくと
も1つを含むことを特徴とする請求項 9 記載の情報処理方法。

【請求項 14】 前記命令は、通知または処理対象情報に対する処理を行な
うための命令を含むことを特徴とする請求項 9 記載の情報処理方法。

【請求項 15】 前記命令は、処理対象情報に対する印刷または保留の少な
くとも1つを行なうための含むことを特徴とする請求項 14 記載の情報処理方法
。

【請求項 16】 前記期限設定工程により設定された期限を過ぎた場合に、
前記設定工程により設定された命令を削除する管理工程を有することを特徴とす
る請求項 10 記載の情報処理方法。

【請求項 17】 処理対象情報の属性を指定する属性指定工程と、
処理対象情報に対する処理を指定する処理指定工程と、
実行すべき命令を設定する設定工程と、
前記属性指定工程により指定された属性の処理対象情報に対して前記処理指定

工程により指定された処理が実行された場合に、前記設定工程より設定された命令を実行する実行工程とを備えるプログラムを記憶したことを特徴とするプログラム記憶可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、情報処理装置及びその方法、及びそのプログラムを記憶した記憶媒体に関するものである。

【0002】

【従来の技術】

従来の情報処理装置においては、エラーが生じた場合に警告を行なったり、印刷が終了した場合にユーザに通知を行なうなどの処理が行われている。

【0003】

【発明が解決しようとする課題】

しかしながら、上述の処理は、装置が固定的に提供する機能であり、ユーザには、その機能を利用するか否かといった選択しかなく、情報の処理に関するいかなる状況において、どのような処理を行なうかということをユーザが自由に設定して実行させることはできなかった。

【0004】

【課題を解決するための手段】

上記課題を解決するために、本発明によれば、情報処理装置に、処理対象情報の属性を指定する属性指定手段と、処理対象情報に対する処理を指定する処理指定手段と、実行すべき命令を設定する設定手段と、前記属性指定手段により指定された属性の処理対象情報に対して前記処理指定手段により指定された処理が実行された場合に、前記設定手段より設定された命令を実行する実行手段とを備える。

【0005】

また、他の態様によれば、情報処理方法に、処理対象情報の属性を指定する属性指定工程と、処理対象情報に対する処理を指定する処理指定工程と、実行すべ

き命令を設定する設定工程と、前記属性指定工程により指定された属性の処理対象情報に対して前記処理指定工程により指定された処理が実行された場合に、前記設定工程より設定された命令を実行する実行工程とを備える。

【0006】

また、他の態様によれば、プログラムを記憶可能な記憶媒体に、処理対象情報の属性を指定する属性指定工程と、処理対象情報に対する処理を指定する処理指定工程と、実行すべき命令を設定する設定工程と、前記属性指定工程により指定された属性の処理対象情報に対して前記処理指定工程により指定された処理が実行された場合に、前記設定工程より設定された命令を実行する実行工程とを備えるプログラムを記憶している。

【0007】

【発明の実施の形態】

以下、図面を用いて本発明の1実施形態を詳細に説明する。

【0008】

図1は、本実施形態の機能構成を示した図であり、ユーザ101、クライアント102、サーバ103、データベース104、デーモン105の関係を示したものである。クライアント102と、サーバ103は同一のデバイス上にあっても良いし、ネットワーク等で接続された別のデバイス上にあっても良い。

【0009】

ユーザ101が何らかの操作をクライアント102に対して行うと、クライアント102はその操作に対応する要求を生成し、サーバー103に送信する。サーバ103は、受け取った要求を解釈して、データベース104とやり取りして、Jobの追加・削除やデータの取得を行うなどの処理を行う。その結果、必要に応じて、対応するHTMLページを作成して、クライアント102に送信する。クライアント102は、受取ったHTMLページを表示し、ユーザ101に新たな操作を促す。

【0010】

データベース104に保存されたJobはデーモン105により監視され、デーモン105は、実行条件が満たされたJobに対応した、印刷、送信、通知などの処理を実行させる。

【0011】

図2は、本実施形態の機能構成の詳細を表した図である。

【0012】

クライアントコンポーネント102は、操作部201及び2つのWeb Browser 202,203から構成される。操作部201を通して行われたユーザの操作は、Web Browser 202,203によって、要求に変換され、サーバコンポーネント103に送信される。

【0013】

サーバコンポーネント103は、Web Server 204、デバイスマネージャ205、Requestマネージャ207、Command解析・処理部208から構成される。クライアントコンポーネント102から送信された要求は、Web Server 204により受信され、デバイスマネージャ205またはRequestマネージャ207に渡される。

【0014】

デバイスマネージャ205は、プリンタ206のステータスを取得したり、制御を行う。Requestマネージャ207は、要求を処理コマンドに変換し、Command解析・処理部208に渡す。

【0015】

Command解析・処理部208では、指示された処理コマンドに対応した処理を実行し、Databaseマネージャ209を介して、データベース104にアクセスし、Jobの追加・更新をする。

【0016】

一方、Daemon105は、一定間隔でDatabaseマネージャ209を介して、データベース104にアクセスし、実行条件が満たされたJobが存在するかどうかチェックし、存在する場合、対応する処理を実行する。

【0017】

図3は、ユーザの操作に対応した情報の流れを説明した図である。

【0018】

クライアントコンポーネント102は、ユーザの操作を入力情報として受け取り、要求を生成して、サーバコンポーネント103に送信する。

【0019】

サーバ コンポーネント103は、受け取った要求を解釈して、データベースマネージャ209を介して、データベース104とやり取りして、Jobの追加・削除やデータの取得を行うなどの処理を行う。また、デバイスマネージャ205を介して、プリンタ206とやり取りをして、プリンタのステータスを取得するなどの処理を行う。その結果、必要に応じて、対応する表示命令（HTMLページ）を作成して、クライアント102に送信する。

【0020】

クライアント102は、受取った表示命令（HTMLページ）をユーザに表示し、新たな操作を促す。

【0021】

データベース104に保存されたJobはデータベースマネージャ209を介して、デーモン105により監視され、実行条件が満たされたJobに対応した、印刷、送信、通知などの処理を実行させる。

【0022】

図4は、ユーザの入力を受け取った後、対応するコマンドの処理が実行されるまでの処理の流れを示した図である。

【0023】

ユーザが操作を行うと、操作部201により対応する操作指示データに変換される。例えば、ユーザがキーボードやマウスを用いて行った操作は、対応するキーコードや、マウス移動情報等の情報処理装置が扱える情報に変換される。

【0024】

その後、上記で変換された操作指示データを受け取ったWeb Browser 202は、Web Server 204への対応する要求に変換し、送信する。例えば、画面に表示されているボタンの1つを選択した場合、「特定のボタンを選択」という要求が作成されることになる。

【0025】

上記要求を受け取ったWeb Server 204は、受け取ったデータをRequest マネージャ207に渡し、Request マネージャ207では要求に対応する処理コマンドに変換

し、Command 解析・処理部208に渡す。例えば、ユーザが印刷を指示するボタンを選択した場合、「印刷を指示するボタンを選択」という要求から、「印刷を指示するボタンを選択に対応する処理実行」という処理コマンドに変換され、実行される。

【0026】

図5は、実際に行われた処理の結果や、プリンタのステータスの変化に対応して、ユーザに情報を表示するまでの処理の流れを示した図である。

【0027】

処理結果を受け取ったRequest マネージャ207は、対応する操作指示データをWeb Server 204に渡す。また、同様にプリンタのステータスの変化を検知したデバイスマネージャ205は、対応する操作指示データをWeb Server 204に渡す。例えば、処理結果として新たな操作画面をユーザに表示する場合、画面に表示する為の情報をHTMLのようなデータに変換することで、実現している。また、プリンタのステータスも同様にして、画面に表示する為の情報をHTMLのようなデータに変換することで、実現している。

【0028】

上記操作指示データを受け取ったWeb Server 204は、Web Browser 202にデータを送信し、Web Browser 202では実際にタッチパネルなどの表示操作部に表示することで、ユーザに示す。

【0029】

図5は、実際に行われた処理の結果や、プリンタのステータスの変化に対応して、ユーザに情報を表示するまでの、処理及びデータの流れを示した図である。

【0030】

処理結果を受け取ったRequest マネージャ207は、対応する操作指示データをWeb Server 204に渡す。また、同様にプリンタのステータスの変化を検知したデバイスマネージャ205は、対応する操作指示データをWeb Server 204に渡す。例えば、処理結果として新たな操作画面をユーザに表示する場合、画面に表示する為の情報をHTMLのようなデータに変換することで、実現している。また、プリンタのステータスも同様にして、画面に表示する為の情報をHTMLのような

データに変換することで、実現している。

【0031】

上記操作指示データを受け取ったWeb Server 204は、Web Browser 202、203にデータを送信し、Web Browser 202では実際にタッチパネルなどの表示操作部に操作画面を表示し、ユーザに示す。Web Browser 203は、プリンタのステータスを表示する。

【0032】

図6は、複数のデバイス（デバイスAと、デバイスBとする）間における、機能の関係を表した図である。ここでは、デバイスAは印刷装置であり、デバイスBは、ユーザのメールやスケジュールを管理するデスクトップ・パソコンとする。デバイスA及びデバイスBには、それぞれクライアントコンポーネント102、サーバコンポーネント103、デーモン105が存在する。但し、ここでは、ユーザがデバイスA側にいるものとして、デバイスBのクライアントコンポーネント102は省略している。

【0033】

ここで、ユーザがデバイスAで操作を行ったとすると、操作部201により、ユーザの操作情報が入力情報として取得され、サーバコンポーネント103のデバイスマネージャ205及びRequestマネージャ207に要求として渡される。

【0034】

デバイスマネージャ205は、プリンタ206のステータスを取得するなどにより、ユーザからの要求に答える。また、Requestマネージャ207は要求の種類により、印刷制御部601によりプリンタを制御したり、更にCommand解析処理部208により、より詳細な情報を解析し、対応する処理を実行させる。

【0035】

その結果、必要に応じて情報の形式を、変換制御部603を用いることで変換したり、制御部602を介して、様々な処理を実行する。ここで、Jobの管理が必要と判断されたならば、Databaseマネージャ209を介して、Database104の追加・更新を行い、表示が必要と判断されたならば、表示制御部604によって表示を行う。また、通知が必要と判断されたならば、通知制御部605によって通知を行う。

【0036】

一方、Daemon 105は定期的にDatabase 104を監視し、実行条件が満足されたJobが存在すれば、対応する処理を実行させる。

【0037】

ここで、デバイスAで処理が完結しないと判断された場合、デバイスBに要求が送信され、デバイスBのRequestマネージャ207が対応する処理を実行させる。例えば、デバイスBのファイルの印刷の場合、印刷部607によって、対応するアプリケーションを用いて印刷処理が実行される。また、Mailアクセスの場合、Mail制御部608によって、対応するアプリケーションを用いて、アクセスされる。この他にも、スケジュール情報の参照など、対応するアプリケーションが用いられて実現される。

【0038】

更に、デバイスBが持つDatabase104に格納されたJobに対しても、アクセスすることができる。

【0039】

図7は、本実施形態に係るシステム構成図である。

【0040】

同図において、ネットワーク701には、種々の装置が接続され、ネットワーク701を介してデータの送受信が行われる。プリンタ702は、ネットワークを介して受信したデータを印刷するとともに、入出力操作部703を有する。入出力操作部703は、ユーザに対して各種表示を行うとともに、ユーザからの指示を受け付ける。スキャナ704、マルチファンクション機器705も同様の入出力操作部を有している。スキャナ704は、紙などに印刷されたデータを光学的に読み取るものであり、マルチファンクション機器705は、プリンタ702とスキャナ704の機能を併せ持っている。パーソナルコンピュータ706は、文書や画像を作成したり、ユーザのメールやスケジュールなどのパーソナルデータを管理する。

【0041】

図8は、本実施形態に係るプリンタの機能ブロック図である。

【0042】

同図において、タッチパネル801は、ユーザに対してプリンタのステータスやメニュー画面の表示を行うとともに、ユーザが表示面に触れることで、メニュー項目の選択や表示されるキーボードによる文字の入力を行うことができる。なお、タッチパネル801に代えて、通常のディスプレイとキーボードを用意してもよい。

【0043】

CPU802は、フローチャートにつき後述する処理手順を含む各種プログラムを実行し、システムバス808により接続された装置各部を制御する。印刷部803は、データを紙などに印刷する。通信部804は、ネットワークを介して、デスクトップやスキャナ、他のプリンタなどと、命令やステータスを含むデータを送受信する。

【0044】

ROM805は、固定的なデータやプログラムを記憶する。RAM806は、データやプログラムを一時記憶する。ハードディスクドライブ(HDD)807は、プログラムやデータを永続的に記憶でき、上述したデータベースとして利用される。システムバス808は、上述した装置各部を接続し、装置間でデータやアドレス、制御信号を送受信する媒体として利用される。フローチャートにつき後述する処理手順を含む各種プログラムは、ROM805に記憶されていてもよいし、HDD807からRAM806へ、処理に先立ってあるいは処理の実行中に必要に応じてロードされるようにしてもよい。

【0045】

図9は、電源投入時の起動順序を示すフローチャートであり、ユーザがマシンの電源を入れた際の起動順序を示している。ステップS901でユーザが電源をonすると、ステップS902でOSが起動し、続いてステップS903でwebサーバが起動する。次にステップS904でクライアントが起動して、最後にステップS905でデーモンが起動する。

【0046】

図10は、クライアントコンポーネントの動作を表すフローチャートである。ク

ライアントが起動すると、ステップS1001でウェブブラウザオブジェクトを作成して、ステップS1002でサーバコンポーネントへ要求を出し、HTMLページ作成を指示する（この時のページは、Opening Screenである）。その要求に対してサーバから送られたHTMLページをステップS1003で表示し、ステップS1004でユーザの入力を待つ。そして、ユーザの入力によって対応する処理を行う。ユーザの要求がexitである場合には、ステップS1006でシステムをシャットダウンして終了する。ユーザの要求がクライアントサイドで処理すべき場合は、ステップS1009で処理し、再びユーザの入力待ち状態になる。サーバサイドで処理すべき場合はステップS1008でサーバコンポーネントへ処理要求を出し、その要求に対して作成されたHTMLページを受取り、表示する。その後、再びユーザの入力待ち状態になる。

【0047】

図11は、クライアントの要求に対するサーバコンポーネントの処理手順を示すフローチャートである。

【0048】

サーバコンポーネントは、ステップS1101でクライアントの要求を取得し、ステップS1102でこの要求を解析し、要求に対応する処理を行う。OpeningScreenを表示する要求に対しては、ステップS1104でプリンタステータスを取得する。Cancel All Jobsに対しては、ステップS1105で、図12につき後述する関数CancelAllPrinterJobs()関数を呼んで、全てのJobをキャンセルする。Cancel/Pause/Restart要求に対しては、ステップS1106でSetPrinterJobsStatus()関数を呼び、Jobを指定されたStatusに設定する。

【0049】

以上のどの要求に対しても、最後にステップS1107で対応するHTMLページを作成し、ステップS1108でクライアントコンポーネントにHTMLページを送信して処理を終える。

【0050】

要求が「login」「logout」であった場合は、図15につき後述する。要求が「Goto Other Device」「Goto Desktop」であった場合は、図16につき後述する。要

求が、「Search」「Help」「Receive Job / Receive Notification」であった場合は、図17につき後述する。要求が、「Print」「Send」「Delete」「Reschedule」「Preview」であった場合は、図22につき後述する。

【0051】

図12は、CancelAllPrinterjobsの処理手順を示すフローチャートである。プロトタイプは、DWORD CancelAllPrinterjobs()である。

【0052】

まず始めに、ステップS1201で、図13につき後述する関数GetAllPrinterJobsを呼び、スプールからJobを取得する。ステップS1202で特定のプリンタをオープンし、ステップS1203で各々のJobのステータスをキャンセルにセットする。その後ステップS1204でプリンタをクローズする。これを、Jobがすべてキャンセルにセットされるまで行う。

【0053】

図13は、GetAllPrinterjobsの処理手順を示すフローチャートである。プロトタイプは、Int GetAllPrinterjobs(pInfo)である。

【0054】

まず、ステップS1301でEnumPrinter()関数を用いて、有効なプリンタ、プリンタサーバ、ドメイン、プリント、プロバイダの列挙を行う。ステップS1302で列挙された対象のうちの一つをオープンし、ステップS1303でEnumJob()関数を用いてスプール中の有効なJobを列挙し、ステップS1304でプリンタをクローズする。この操作を全対象について行う。

【0055】

図14は、SetPrinterJobStatusの処理手順を示すフローチャートである。プロトタイプは、Int SetPrinterJobStatus(pPrinterName,JobId,Status)である。

【0056】

まず、ステップS1401でGetJob関数を用いて、スプールから指定されたJobを取得する。ステップS1402で、そのJobが存在する特定のプリンタをオープンし、ステップS1403でJobのステータスを指定された値にセットする。ステップS1404で最後にプリンタをクローズして終わる。

【0057】

図15はクライアントの要求が、「login」「logout」であった場合のサーバの処理手順を示すフローチャートである。

【0058】

要求がloginであったときは、まずステップS1502でlogin verbとリモートデバイスリストを取得し、ステップS1503でユーザ名を取得する。ステップS1504でGetUserId()を呼び、そのユーザに有効なJobリストを取得し、ステップS1505で対応するHTMLページを作成すると同時に、ステップS1506でその隠れ領域にユーザとJobの詳細を保存する。

【0059】

一方、Logout要求の場合には、ステップS1507でリモートデバイスリストが空かを調べ、空のときは、ステップS1510で、ステータスに従ってOpeningScreenページを作成する。空でないときは、リストが空になるまで、ステップS1508で次のデバイスにlogin verb=Disconnectでログインし、ステップS1509でそのデバイスをリモートデバイスリストから削除することを繰り返す。すなわち、リモートデバイスリスト中の全てのデバイスをログアウトする。リストが空になったら、OpeningScreenページを作成する。

【0060】

Login、logoutどちらの場合も、最後にステップS1511で、作成したページをクライアントコンポーネントに送信して終わる。

【0061】

図16はクライアントの要求が、「Goto Other Device」「Goto Desktop」であった場合のサーバの処理手順を示すフローチャートである。

【0062】

要求がGoto Other Deviceの場合、ステップS1602でデバイス名とデバイスアドレスを取得する。Goto Desktopの場合は、ステップS1603でデスクトップアドレスを取得する。その後、ステップS1604でそのデバイスをリモートデバイスリストに追加し、ステップS1605でlogin verbを作成する。login verbがDevice or Desktop、Print、Send、Delete、Rescheduleの場合、ステップS1607でLogin verb

によって、デバイスへログインし、Device or Desktopの場合は、ステップ S1608でオープニングスクリーンのHTMLページを作成し、Print、Send、Delete、Reschedule の場合は、ステップ S1609、1610、1611、1612で、それぞれPrintJob、Send Job、Delete Job、Reschedule JobのスクリーンのHTMLページを作成する。また、login verbがDisconnectの場合、ステップS1613でデバイスからログアウトし、ステップS1614でそのデバイスをリモートデバイスリストより削除し、ステップ S1615でオープニングスクリーンのHTMLページを作成する。最後にステップS1616で、クライアントステップコンポーネントにページを送信して終る。

【0063】

図17はクライアントの要求が、「Search」「Help」「Receive Job / Receive Notification」であった場合のサーバの処理手順を示すフローチャートである。

【0064】

Search要求の場合、ステップS1702で検索パラメータを取得し、ステップS1703で検索を実行する。ステップS1704で、その実行結果から検索結果のHTMLページを作成する。Help要求の場合、ステップS1705でヘルプコンテキストパラメータを取得し、ステップS1706で対応するヘルプスクリーンHTMLページを作成する。どちらの要求の場合も、最後にステップS1707で、作成したHTMLページをクライアントコンポーネントに送信して終了する。

【0065】

要求の種類が、Receive Job/Notificationの場合、ステップS1708でデータバッファを読み、ステップS1709で、そのバッファを、Receive Jobであれば、図18につき説明するIdAnalyze()関数へ、Receive Notificationであれば、図20につき説明するIdAnalyzeNotifyへ渡して終了する。

【0066】

図18は、IdAnalyzeの処理手順を示すフローチャートである。ここでは、他のデバイスから行データとして渡されたJobを解釈する。このJobはDBに加え、データモジュールによって実行される。

【0067】

まず、ステップS1801でDemarshal関数を呼び、demarshalを行い、バッファオブジェクトを有効にする。次に、ステップS1802で、圧縮データをUnformatData関数を用いて復号、展開をおこなうことによって、行単位でデータが有効になる。ステップS1803で再びDemarshal関数を呼び、メッセージオブジェクトを有効にする。最後にステップS1804で図19につき後述するReceiveJob関数(DUMMY)を呼び、JobをDBに追加する。

【0068】

図19は、DUMMYの処理手順を示すフローチャートである。このモジュールはSendJobにより送られた行バッファを受取る、ISAPI拡張Receiveモジュールのダミーである。CGIアプリケーションへ渡される。

【0069】

実際には、ステップS1901でSendJobにより送られた行バッファ中のこれらの関数を読み、ステップS1902でテンポラリファイルをダンプする。このファイル名は、ステップS1903でCGIモジュールのIDExtn.Exeへ渡される。これは、ステップS1904でファイル名を読み、データを取得する。ステップS1905でCGIアプリケーションへデータを送るために、このモジュールはinternet client callを使う。

【0070】

図20は、IdAnalyzeNotifyの処理手順を示すフローチャートである。他の同種のデバイスから行データとして渡されたNotificationを解釈する。このnotificationを処理に加えnotificationはデーモンモジュールによって実行される。

【0071】

実際には、ステップS2001で、Demarshal関数を呼び、demarshalを行い、バッファオブジェクトを有効にする。つぎに、ステップS2002で圧縮データをUnformatData関数を用いて復号、展開をおこなうことによって、行単位でデータが有効になる。ステップS2003で、再びDemarshal関数を呼び、メッセージオブジェクトを有効にする。最後に、ステップS2004で、図21につき後述するRecieveHttpNotification関数を呼び、JobをDBに追加する。

【0072】

図21は、`RecieveHttpNotification`の処理手順を示すフローチャートである。
プロトタイプは、`Void CIDAction::RecieveHttpNotification(poNotify)`である
。

【0073】

この関数はHTTPプロトコルによって送られたnotificationを受取る関数である。
。デバイスが要求された通知方法をサポートしていないなくても、中間デバイスを利用して最終的には目的のデバイスに通知することが出来る。

【0074】

ステップS2101で、`Notification`のレシーバ名とアドレス、通知方法を調べ、
ステップS2102で、このデバイスがnotificationを受取る目的のデバイスか、他の
デバイスへnotificationをpassする中間点として機能するのみかを調べる。

【0075】

このデバイスが最終目的値である場合、ステップS2103でテンポラリファイルを作成し、ステップS2104で、その中へnotificationメッセージを書き込む。また、ステップS2105でウインドウズディレクトリ上のfixedファイル”pollfile”（デーモンがpollするファイル）を作成、上書きする。ステップS2107で、このファイルの中でnotificationモジュール（メッセージを表示したり、作成したりするモジュール）のコマンドラインパラメータを指定し、ステップS2107で指定された通知方法で通知する。

【0076】

このデバイスが中間デバイスである場合、ステップS2108で、このデバイスにnotificationを発行する能力があるかを確認し、あれば、ステップS2110で、指定された方法で指定されたデバイスへnotificationを送信する。このデバイスにnotificationを発行する能力がない場合、ステップS2112で、HTTPプロトコルを介して他の中間点へnotificationを送信する。

【0077】

図22は、クライアントの要求が、「Print」「Send」「Delete」「Reschedule」 「Preview」であった場合のサーバの処理手順を示すフローチャートである。

【0078】

これらの5つの要求の場合、まずははじめに、ステップS2201で、ユーザID、JobID、ActionIDを取得する。

【0079】

Print要求の場合、ステップS2203で、印刷時刻と場所を取得し、ステップS2204でnotificationリストを取得する。ステップS2205で関数PrintJob()を呼びプリントを実行する。

【0080】

Send要求の場合、ステップS2206で送信時刻と場所を取得し、ステップS2207でnotificationリストを取得する。ステップS2208で関数SendJob()を呼び送信を実行する。

【0081】

Delete要求の場合、ステップS2209でCancelAction()関数を呼び、Jobを削除する。

【0082】

Reschedule要求の場合、ステップS2210でRescheduleAction関数を呼び、スケジュール変更を行う。

【0083】

Preview要求の場合、ステップS2211でformat conversion libraryを用いて、JobをJPGやGIFフォーマットに変換する。

【0084】

以上の要求全てにおいて、ステップS2212で、対応するHTMLページを作成して、ステップS2213でクライアントコンポーネントにHTMLページを送信して終了する。

【0085】

図23は、PrintJobの処理手順を示すフローチャートである。プロトタイプは、
`void CIDAction::PrintJob(ActionId, NotifyList)`である。入力パラメータは、指定されたActionのID、ユーザのNotification設定リストであり、Jobを、今すぐにこのデバイスで印刷する。

【0086】

ActionIDがPrint/Holdのとき、Actionの属性（Print時刻、Notificationなど）は、現時刻と新しいNotificationに修正される。ActionIDがSendのとき、DB中に新しいActionが作成される。これは、そのこのユーザがHoldしている、このJobに関係する他のPrintActionが存在するかどうかに依存する。Jobがプリントされた後は、ActionStatusは”done”となり、このActionはユーザのHistoryの一部になる。

【0087】

まず、ステップS2301で、レコードを引き出して、Actionの詳細をチェックする。ステップS2302で、古いActionがHoldされているかチェックし、ステップS2303でこの情報を保存する。ステップS2304でこのActionに必要なJobDataを取得する。ステップS2305で、データベース中のエントリを“印刷中”となるよう修正する。これは、daemonモジュールがこのレコードを実行しないようにするためにある。

【0088】

ステップS2307、2308で、新しいレコードを作成するかまたは、存在するレコードを修正する。たとえば、前のActionがPrint/Holdであったならそのレコードは修正され、前のActionがSendであったなら、そのJobに対してそのユーザがHold中のPrintActionがないときには新しいレコードが作成される。new/old Action IDが返される（条件に従う）。

【0089】

ステップS2309で、無条件で、Notification StatusはFalseになる。ステップS2310で、このnewActionIDにnotificationを加えてユーザータイプを”Don't care”とする（すなわち、送信者も受信者もいない）。

【0090】

ステップS2311で、Spooler API Callsを使って印刷するために、プリントキーへJobを送る。ステップS2312で“印刷が成功”または“終了”とエントリを修正する。ステップS2313で、印刷のActionStatusに従ってNotificationStatusをセットする。

【0091】

図24は、SendJobの処理手順を示すフローチャートである。プロトタイプは、`void CIDAction::SendJob(bsendjobToDatabase, ActionId, roJobMessage)`である。

【0092】

指定された“存在する”Job、または指定されたレシーバ・デバイスリストへ新しいJobを送信する。送信によって、1つのデバイスに対して、1つのコピーが送られる。この場合、複数のレシーバがいる場合には、Jobのコピーを共有していることになる。他の全てのレシーバの情報も送られるので、他のデバイスからJobにアクセスすることも可能である。JobはHTTPプロトコルで送信される。

【0093】

実際には、ステップS2401で、`ActionId`が存在するJobか、新規のJobかをチェックする。`ActionId`が存在する場合には、ステップS2402で、`Action`の属性を取得して、ステップS2403で、その`Action`タイプが`Hold`であるかチェックする。`Hold Job`であるとき、ステップS2404で、`Hold Job`がアクセスされたという情報をストアする。

【0094】

次に、この情報がDBへ行くと指定されているときには、ステップS2406で、`SendJob()`を呼び、Jobに関する全ての情報（センダ、レシーバなど）をDBにストアする。その後、ステップS2407で“今”送信するかどうか、時間をチェックし、今送らないときには、そのままに終了して、デーモンモジュールが送信を実行してくれるのを待つ。今送るときには、ステップS2408で、送信を実行して終了する。

【0095】

図25は、`SendJob()`による送信の実行手順を示すフローチャートである。

【0096】

まず、`EditAction()`を呼び、`ActionStatus`を`active`にセットする。次にレシーバのリストを処理し、ユーザ名、デバイス名、デバイスのURLアドレスを取得する。レシーバデバイスに対応するようにフォーマット変換を行う。これは適当なイメージ変換ライブラリを用いて実行される。メッセージデータをバイト列に変

換し、さらに圧縮、符号化を行う。このデータをHTTPプロトコルを用いて送信し、`ActionStatus`を`done`に設定する。さらに`Action`をユーザのヒストリの一部として追加し、送信状況によって`notification status`を`success/fail`にセットする。Jobが`Held List`からアクセスされたjobであることを示す`Status`をセットする（ステップS2501～2511）。

【0097】

図26は、`CancelJob`の処理手順を示すフローチャートである。プロトタイプは、`Void CIDAction::CancelJob(dwActionId)`である。

【0098】

この関数は指定された`Action`をキャンセルするものである。具体的には、ステップS2601で、`Action`の属性を取得して、`CancelAction()`関数を読んで、`Action`をキャンセルとマークする。もし、`Action`が`Hold`リストの中にある場合には、ステップS2603で、ユーザのヒストリの一部に追加し、ユーザのヒストリの一部になっている場合には、ステップS2604でエントリを削除する。`Action`が`Hold`である場合には、ステップS2606で、`notification condition`は“`Hold Job`にアクセスしている”と設定する。ユーザがこのデバイス上でのレシーバであるときには、そのまま終了し、他のデバイス上にいる場合には、ステップS2608で、このデバイスへログインさせ、ステップS2609で、キャンセル`Action`に対しての詳細を設定し、ステップS2610で、実際のデバイスにこの情報を送るために`SendJob()`関数を呼び、送信を実行する。このことにより、`Action`のキャンセルが実行される。

【0099】

図27は、`SendNotification`の処理手順を示すフローチャートである。プロトタイプは、`Void CIDAction::SendNotification(dwActionId)`である。

【0100】

この関数は、実際の通知を行う関数である。`Popup`などの通知のタイプは、`http`を送信することによって通知を行う。電話やメールの場合は、適当なモジュールを呼ぶことによって通知する。サポートされたいない方法が指定された場合には、指定された中間デバイスへ`notification`が送信されて、再送される、この機

能は"now"にnotificationを送信しなければならないときにのみ行われる。この場合、NotificationStatusをDoneにして、デーモンモジュールがpollしないようとする。

【0101】

実際には、まず、ステップS2701で通知方法を調べ、その方法がpopup windowやvoice messageのときには、ステップS2703で、httpプロトコルでデータが送信される。方法が電話などのときには、ステップS2705で、そのデバイスに通知能力があるかを調べ、その能力がある時には、ステップS2704で、その指定された方法で通知を行う。デバイスに能力がないときには、ステップS2706で中間点を調べる。この中間点では指定された発行のメカニズムをサポートしている。ステップS2707で、通知のメッセージとその詳細をHTTPプロトコルで中間デバイスに送信する。この方法でメール、Fax、ポケベルなどにも対応できる。

【0102】

図28は、サーバに対する入力情報と処理との対応関係を示す図である。それに対応する処理をまとめたものである。

【0103】

ここで、Print Job、Send Job、Cancel Job、Send Notification、Receive HTTP Notificationは、制御部602が司る。このモジュールは、プリンタのバックエンドのメインとなるモジュールであり、「ここで、今すぐにJobを印刷する。」「ここで、後でJobを印刷する。」「違う場所で、今すぐにJobを印刷する。」「違う場所で、後でJobを印刷する。」「今すぐに、または後で、他デバイスにJobを送信する。」「スケジュールを変更する。」「JobのActionをキャンセルする。」「通知のタイプを設定する（音声、ポップアップウィンドウ、電話、E-mail、Fax）。」「通知を受取る。」「デバイスのステータスを取得または設定する。」といった動作を実現する。

【0104】

このモジュールは、DBとその呼び出しがそれを用いるモジュール（GUIモジュールやDaemonモジュールなど）にtransparentとなる形でDBモジュールに覆い被さっている。

【0105】

図29は、デーモン105の起動からの処理手順を示すフローチャートである。

【0106】

デーモンモジュール105は、絶えずDBとファイルをポーリングして、後で実行されるように登録されているActionや操作を実行する。これらの全ての場合、エントリはDB中に作成される。

【0107】

まず、ステップS2901で、無限ループ中でポーリングを行うスレッドを作成し、“今”実行しなければならないJobをpollする。DB中にJob(Action)を見ると、そのActionタイプを調べる(ステップS2902～2905)。

【0108】

ActionがPrintである場合には、ステップS2906で、PrintJob()関数を呼び、印刷を実行する。ActionがSendである場合には、ステップS2908で、ConstructSendMessage()を呼び、メッセージを作成した後、ステップS2909でSendJob()関数を呼び、送信を実行する。

【0109】

ActionがHoldである場合には、ステップS2907でCancelJob()関数を呼び、Jobをキャンセルする。

【0110】

これらの処理の後、ステップS2910でnotificationを調べる。

【0111】

図30はデーモンモジュールのNotification処理手順を示すフローチャートである。

【0112】

まず、ステップS3001で、このデバイスから発行しなければならない通知が存在するかを、関数PollNotify()を呼ぶことによって調べる。発行すべきnotificationが存在する場合には、ステップS3002で、SendNotification()関数を読んで通知を発行する。Notificationがなければ、ステップS3003で、アプリケーションファイルの存在をチェックする。

【0113】

ファイルが存在する場合には、ステップS3005で、このファイルからコマンドラインパラメータを読んだ後、このファイルを削除する。ShowNotify.Exeモジュールを起動し、コマンドラインパラメータをpassする。

【0114】

最後に、PullPrintの場合には、ステップS3007でPrint0cx.EXEファイルを起動し、Scheduleの場合には、ステップS3008でDataDumo.Exeファイルを起動する。

【0115】

図31はデーモンが扱うjobの種類と対応する処理との関係を表した図である。

【0116】

図32は、PrintPrinterJobの処理手順を示すフローチャートである。プロトタイプは、Int PrintPrinterJob(pPrinterName,pDocumentName,pVoid,BufSize)である。

【0117】

この関数はプリンタにJobがスプールされていることをプリントする関数である。実際には、指定されたプリンタをオープンし、そのプリンタにJobがスプールされていることを通知し、そのプリンタにデータを書き込み、プリンタをクローズする（ステップS3201～3204）。

【0118】

図33は、Sendjobの処理手順を示すフローチャートである。

【0119】

ステップS3301で、送られるジョブが新規か、データベース中に既存のものかをチェックする。新規のジョブである場合、ステップS3303で、その詳細をデータベースに加える。ステップS3311で、ジョブのSender・Receiverも同様にチェックする。それらがデータベースに存在する場合、ステップS3312で新規ActionはユーザIDに加えられ、データベース中に存在しないものに対しては、ステップS3313で新規エントリを加える。

【0120】

既存のジョブが送られる場合にはSenderへのActionをチェックする。Actionが

Sendで、ステータスがTodoまたはActiveである場合、ActionのWhenTodoタイムをJobMessageが作成された新しい時間に変更する。ステップS3313で、ActionがSendで、既に実行されている場合には、新しいActionをデータベースに加える。ActionがSendでない場合、このジョブにエントリされたActionが、データベースに作成される。

【0121】

全ての場合において、ジョブのReceiverの存在がチェックされ、もし、Receiverがデータベースに存在しない場合には、新規エントリが作成される。ActionがHoldで、ActionStatusがTodoのとき、ActionをSendに変更しWhenTodoタイムをJobMessageが作成された新しい時間に変更する。ActionStatusがTodoでないときには、ActionがSendである新しいレコードを作成し、WhenTodoTimeをJobMessageに指定された新しい時間に設定する。

【0122】

図34は、ReceivejobDataの処理手順を示すフローチャートである。

【0123】

まず、ステップS3400で、"Deny Jobs From Person/Device"というチェック項目を見て、ここに設定されている送信元からのジョブの場合、ステップS3417で、ジョブの受け取りを拒否する。この項目は、管理者によって設定される。

【0124】

図134は、ジョブの拒絶の設定のための設定画面を示す図である。同図において、"Deny Jobs From Person"には、拒否すべきユーザの識別子を設定する。"Deny Jobs From Device"には、拒否すべき装置を設定する。更に、図77につき後述する命令の設定において説明するように、拒否する。

【0125】

有効期限や拒否すべき処理の種類を設定するようにしてもよい。以上のようにして設定されたチェック項目をみて、送信元のユーザあるいは装置からのJobの受け取りの許可・不許可を判断し、許可されていれば、受け取り処理を行う。

【0126】

受け取られるジョブは、新しいジョブである場合と、既存のジョブである場合

の二つの場合が想定される。

【0127】

新しいジョブが受け取られる場合、ステップS3402で、新しいエントリをジョブの詳細に加える。Senderが存在しない場合は、ステップS3404でSenderのエントリを加える。ステップS3405でAction、Notificationsも同様に加える。Senderが存在する場合、ステップS3406でNotificationsを加える。もしもReceiverが存在しない場合にはステップS3408で追加する。ステップS3409でReceiverのActionも追加する。

【0128】

既存のジョブが受け取られる場合、ジョブのReceiverをチェックする。もし、Receiverが存在し、受け取られたのと同じActionを持つ時には、前のもののActionステータスはキャンセルとされ、新規にレコードが作成される。Receiverに関するActionが受け取られたのと違うActionを持つ時には、新規エントリがReceiverに関して加えられる。場合によってはNotificationsも加えられる。User typeが異なる場合を除き、ActionステータスはTodoに設定される。

【0129】

図35は、GetListOfJobAttributeの処理手順を示すフローチャートである。

【0130】

指定された場所でのユーザの全ジョブをチェックし、そのユーザのジョブの全詳細を返す。ジョブはHeldjobs、またはHistoryjobsの2つである。データベースはUserIDごとにジョブをサーチし、全てのジョブがピックアップされてリストに加えられる。このリストが返される(ステップS3501～3507)。

【0131】

図36は、PollJobの処理手順を示すフローチャートである。

【0132】

UserタイプがReceiverで、ActionStatusがTodoで、WhenTodoが現システム時間よりも短いを検索する(ステップS3601～3605)。Jobが存在したら、Jobの詳細を取得し、ActionSetにデータを設定して、TRUEを返す(ステップS3606～3608)。Jobが存在しないときには、FALSEを返す(ステップS3609)。

【0133】

図37は、`PollNotify`の処理手順を示すフローチャートである。

【0134】

`CanNotify`がTRUEで、`WhetherToNotify`がTRUEで、`WhenToNotify`が現システム時間よりも短い最初のレコードを検索する（ステップS3701～3705）。レコードが存在するときには、`notification`の詳細を取得し、`NotificationSet`にデータを設定して、TRUEを返す（ステップS3706～3708）。レコードが存在しないときは、FALSEを返す（ステップS3709）。

【0135】

図38は、`CancelAction`の処理手順を示すフローチャートである。

【0136】

まず、指定された`ActionID`を持つレコードを検索する。該当レコードが存在すれば、同じ`ActionID`をもつ`Job`を検索し、そのサイズをチェックする（ステップS3801～3804）。そのサイズが管理者が設定する`JobSizeLimit`よりも小さい場合には、ステップS3805で、`ActionStatus`を調べる。この`ActionStatus`が`Done`であるか、ジョブのサイズが`JobSizeLimit`以上の場合は、`ActionStatus`を`Deleted`に変更し、そのレコードを削除する（ステップS3807～3808）。それ以外のときは、ステップS3806で、`ActionStatus`をキャンセルとする。そして、ステップS3809で、`ActionID`に対応する全ての`Notification`を削除する。

【0137】

図39は、`EditAction`の処理手順を示すフローチャートである。

【0138】

指定された`ActionID`を持つレコードを検索し、該当レコードが見つかったら、`Action`ステータスを受け取ったパラメータに変換する（ステップS3901～3903）。

【0139】

図40は、`EditAction`の処理手順を示すフローチャートである。図39につき説明したものとは、プロトタイプが異なる。

【0140】

指定されたActionIDを持つレコードを検索し、該当レコードのActionがAction IDのActionと一致し、かつ、Actionステータスが保留である場合に、Actionステータスを受け取ったパラメータに変換する（ステップS4001～4006）。該当レコードがない場合には、ステップS4004で、新規Actionを追加する。

【0141】

図41は、RescheduleActionの処理手順を示すフローチャートである。

【0142】

指定されたActionIDを持つレコードを検索する。該当レコードが見つかり、Action Statusがdoneでなければ、WhenTodoを受け取ったNewTimeに変換する（ステップS4101～4104）。

【0143】

図42は、RescheduleActionの処理手順を示すフローチャートである。図41につき説明したものとは、プロトタイプが異なる。

【0144】

ステップS4201で、指定されたActionIDを持つレコードを検索する。該当レコードが見つかり、UserTypeが一致していれば、Actionステータスをキャンセルに変更し、ActionステータスがTodoである新規レコードを追加する（ステップS4201～4204）。該当レコードが見つからない場合、ユーザの存在をチェックする。ユーザが見つかればUserIDをpickし、見つからない場合は新規ユーザを追加してからUserIDをpickする。このUserIDに一致する新規Actionを追加する（ステップS4206～4209）。

【0145】

図43は、RescheduleActionの処理手順を示す図である。図41、42につき説明したものとは、プロトタイプが異なる。

【0146】

指定されたActionIDを持つレコードを検索する。該当レコードが見つかったら、ActionステータスとActionをチェックする。ActionがActionIDと一致し、かつActionステータスが保留で未了の場合、WhenTodoタイムを新しい時間に変更する

。 ActionIDの全Notificationを削除し、新規Notificationを追加する(ステップS4301～4302、4304～4308)。該当レコードが見つからない場合、ステップS4303で、ActionステータスがTodoである新規Actionを追加し、WhenTodoタイムを新しい時間に変更する。

【0147】

図44は、AddNotificationの処理手順を示すフローチャートである。

【0148】

Notificationに新規レコードを追加する。パラメータの総合情報(GeneratedFrom parameter)をチェックして、これに関する新規エントリを作成する。

【0149】

図45は、AddInstructionの処理手順を示すフローチャートである。

【0150】

ReceiveJobまたは、SendJobが発生する時に、Instructionの設定が呼び起こされる。まず、ステップS4501で、データベース上に指定されたUserIdの存在を確認し、存在しないときには処理を終了する。存在するときには、ステップS4502で、Instructionリストからレコードを取得し、ステップS4503で、このUserに新しいinstructionを追加する。

【0151】

Instructionは、データベースのInstructionテーブルにInstructionを追加する。関連するActionと、Instructionが満たされて与えられたNotificationを、Actionテーブル及び Notificationテーブルに反映する(ステップS4504～4506)。

【0152】

Receiveジョブ・Sendジョブいずれの場合においても、これらを生成する Instructionが満たされた時には、Actionテーブル及び Notificationテーブルに新規 Actionと新規 Notificationを生成するテンプレートとして作用するレコードを追加する。

【0153】

図46は、GetInstructionの処理手順を示すフローチャートである。

【0154】

この関数は、ReceiveJobDataとSendJobDataによって呼ばれる。まず、ステップS4601で、Instructionテーブルの全レコードにつき、ActionタイプがReceiveまたはSendであるかをチェックし、指定されたActionタイプの全命令を取得する。ステップS4602で、JobMessage中の詳細（命令として設定されている値）をチェックし、これらの値を用いて、instructionが実行できるか出来ないかを調べる。Instructionの有効性のチェックのために、check関数（後述）を用いる。Instructionが有効である場合には、ステップS4603でそのIdをリストに加え、ステップS4603で、全命令について処理が終了したなら、最後に有効なInstructionIdのリストを返す。

【0155】

図47は、ExecutionInstructionの処理手順を示すフローチャートである。この関数は、ReceiveJobDataとSendJobDataによって呼ばれる。ActionテーブルとNotificationテーブルのレコードを探す。これらのレコードは、Instructionリストの各Instructionに従って実行され、そのコピーをtodoフラグ(TRUEにセット)を付けて各テーブルに作成する（ステップS4701～4708）。

【0156】

instructionによってセットされたdummy actionがあるかActionテーブルをサーチする。これらのdummy actionsに基づいて新しい適当なレコードが追加され、actionが実行される。

【0157】

Instructionによってセットされたdummy notificationがあるかサーチする。これらのdummy notificationに基づいて新しい適当なレコードが追加され、notificationが実行される。

【0158】

図89は、ResetInstructionsの処理手順を示すフローチャートである。プロトタイプは、Void ResetInstructions(dwUserId,pInstructionList)である。

【0159】

まず、指定されたUserIdの全ての命令を検索し、得られた命令と、その命令に

対応する全てのAction,Notificationを削除する(ステップS8901～8905)。その後、命令リストで渡された新しい命令を追加する(ステップS8906)。

【0160】

図90は、DeleteAllInstructionsの処理手順を示すフローチャートである。プロトタイプは、Void DeleteAllInstructions(dwUserId)である。

【0161】

指定されたUserIdの全ての命令を検索し、検索された命令と、その命令に対応する全てのAction,Notificationを削除する(ステップS9001～9005)。

【0162】

図91は、DeleteInstructionsの処理手順を示すフローチャートである。プロトタイプは、Void DeleteInstructions(dwInstructionId)である。

【0163】

指定されたInstructionIdの命令を検索し、検索された命令と、その命令に対応するAction,Notificationを削除する(ステップS9101～9104)。

【0164】

図92は、GetInstructionDetailsの処理手順を示すフローチャートである。プロトタイプは、Void GetInstructionDetails(dwUserId)である。

【0165】

指定されたUserIdの命令を検索し、検索された命令と、その命令に対応するAction, Notificationの詳細をリストにして返す(ステップS9201～9208)。

【0166】

図93は、Checkの処理手順を示すフローチャートである。プロトタイプは、Bool Check(nJobChk, JobNameConnector, nDeviceChk, DeviceConnector, nPersonChk, PerconConnector, nTimeChk, TimeConnector, nStatusChk, nStatusCode)である。

【0167】

この関数は、GetInstructionIds()の中で呼ばれる。命令が実行可能、有効であるかをチェックし、有効ならばTRUE、そうでなければFALSEを返す(ステップS9301～9303)。

【0168】

図94は、GetJobDataの処理手順を示すフローチャートである。プロトタイプは、`Void GetJobData(dwJobId,poJobMessage)`である。

【0169】

指定されたJobIdのJobを検索し、Jobの詳細をJobMessageに設定する(ステップS9401～9403)。

【0170】

図95は、GetJobAttributesの処理手順を示すフローチャートである。プロトタイプは、`Void GetJobAttributes(dwJobId,poJobMessage)`である。

【0171】

指定されたJobIdのJobを検索し、Jobの詳細をJobMessageに設定する。またJobのSenderとReceiverの詳細と、関係するActionの詳細を検索し、その詳細をJobMessageに設定する(ステップS9501～9505)。

【0172】

図96は、GetActionAttributesの処理手順を示すフローチャートである。プロトタイプは、`Void GetActionAttributes(dwActionId,poActionMessage)`である。

【0173】

指定されたActionIdのActionを検索し、Actionの詳細をActionMessageに設定する。またJobのSenderとRecieverのユーザ名、アドレスも、SenderとReceiverのActionMessageに設定する(ステップS9601～9605)。

【0174】

図97は、DeleteJobの処理手順を示すフローチャートである。プロトタイプは、`Void DeleteJob(dwUsrId,dwJobId)`である。

【0175】

ステップS9701で、指定されたJobIdをもつ、指定されたUserIdのActionを検索する。アクションが存在すれば、ステップS9703で、ActionStatusのタイプをチェックする。もしActionStatusがNot Doneであれば、ステップS9704で、ActionStatusをActionCancelに変更し、実行時間を現時刻に設定する。

【0176】

もしActionStatusがすでにDoneまたはCancelである場合には、ステップS9705で、そのActionと、関係する全てのNotificationを削除する。ステップS9706で、指定されたJobIdをもつ指定されたUserId以外のUserIdのActionがあるかをチェックし、なければステップS9707で、指定されたJobIdをもつ、指定されたUserIdのAction全てが既にDoneまたはCanceledであるかをチェックし、他のユーザのActionがなく、Action全てが既にDoneまたはCanceledである場合には、ステップS9708で、JobテーブルからこのJobのレコードを削除する。

【0177】

図98は、GetNotificationListの処理手順を示すフローチャートである。プロトタイプは、Void GetNotificationList(nGeneratedFrom,dwId)である。

【0178】

指定された生成場所 (Action / Instructionのどちらか) から生成され、指定されたIdをもつ全てのNotificationを検索し、その詳細をリストに加えてリストを返す(ステップS9801～9804)。

【0179】

図99は、SetNotificationConditionの処理手順を示すフローチャートである。プロトタイプは、Void SetNotificationCondition(nGenetaredFrom,dwId,nResult)である。

【0180】

指定場所(nGenerateFrom)から生成され、指定されたIdをもつNotificationを検索し、指定された値(nResult)で作成されていれば、WhetherToNotifyをTRUEに、指定された値で作成されていなければFALSEに、セットする(ステップS9901～9905)。

【0181】

図100は、SetNotificationConditionの処理手順を示すフローチャートである。プロトタイプは、Void SetNotificationCondition(nGenetaredFrom,dwId,bCanNotifyFlag,nDummy)である。

【0182】

指定場所から生成された、指定NotificationIdを持つNotificationを検索し、CanNotifyFlagを指定値(bCanNotifyFlag)にセットする(ステップS10001～10003)

【0183】

図101は、SetNotificationStatusの処理手順を示すフローチャートである。プロトタイプは、Void SetNotificationStatus(dwNotificationId, bNotificationStatus)である。

【0184】

指定されたNotificationIdをもつNotificationを検索し、Notification Statusを指定値にセットする。(ステップS10101～10103)。

【0185】

図102は、ClearHistoryの処理手順を示すフローチャートである。プロトタイプは、Void ClearHistory()である。

【0186】

まず、ステップS10201で、Action Statusを調べ、Done,Cancelled,Ignored,DeletedのActionを検索する。得られたActionの1つ1つについて、Action StatusがDoneのJobのときは、ステップS10204で、実行時間をチェックする。それ以外のときは、ステップS10205で、WhenTodo時間をチェックする。ステップS10206で、これらの時間と履歴(History)の保持制限時間とを比較する。このようなジョブに対する制限は、管理者によって設定されるものである。図133に、このための設定画面の例を示す。ここでは、1箇月(1 Month)に設定されている。比較の結果、実行時刻あるいは実行予定時刻から保持制限時間以上経過していた場合、ステップS10207で、ActionとJobは削除される。

【0187】

図103は、DeleteAllPublicInfoJobsの処理手順を示すフローチャートである。プロトタイプは、Void DeleteAllPublicInfoJobs()である。

【0188】

PublicInfo Jobを全て検索し、ActionStatusをActionCancelにセットする(ス

ステップS10304～10304)。

【0189】

図104は、PrintPublicInfoの処理手順を示すフローチャートである。プロトタイプは、Void PrintPublicInfo(dwActionId)である。

【0190】

指定されたActionIdのActionをサーチし、PublicInfoJobがあるかどうかチェックする。存在するときには新しいActionを追加する。このActionをPublicInfoで、ActionがPrint、WhenTodoが現時刻、ActionStatusがTodoであるようなActionに設定する(ステップS10401～10407)。

【0191】

図105は、SearchDataの処理手順を示すフローチャートである。プロトタイプは、Void SearchData(dwUserId,poSearchData)である。

【0192】

指定されたユーザの全Jobリストを取得し、その詳細とSearch Dataオブジェクト中の詳細を比較する。詳細がマッチしたら、新リストへのノードを追加し、最後に新リストを返す(ステップS10501～10507)。

【0193】

図48はデスクトップのサーバコンポーネントのクライアントの要求に対する処理手順を示すフローチャートである。

【0194】

サーバコンポーネントは、ステップS4801で、クライアントの要求を取得し、対応する処理を行う。要求がloginであったときは、まずlogin verbとリモートデバイスリストを取得し、ユーザパスワードを取得する(ステップS4803～4804)。その後ユーザに対応するHTMLページを作成し、ユーザパスワードを保存して、クライアントコンポーネントに送信して処理を終える(ステップS4805～4806、4814)。

【0195】

Print a File要求の場合には、ユーザパスワードと印刷するファイル名を取得する。外部のアプリケーション呼び出し、このアプリケーションが用いる印刷用フ

ファイルへ指定されたファイル名のファイルを印刷する。この出力用印刷ファイルをメッセージオブジェクトに変換し、リモートデバイスの始めのデバイスへJobを送信する。最後に対応するHTMLページを作成し、クライアントコンポーネントに送信して処理を終える(ステップS4807~4813、4814)。クライアントの要求が、「Read a Mail」「Print a Mail」であった場合は、図49につき後述する。クライアントの要求が、「View Schedule」「Print Schedule」であった場合は、図50につき後述する。クライアントの要求が、「Goto Other Device」であった場合は、図51につき後述する。クライアントの要求が、「logout」「help」であった場合は、図52につき後述する。

【0196】

図49は、メールに関する要求の処理手順を示すフローチャートである。これは、クライアントの要求が、「Read a Mail」「Print a Mail」であった場合の手順である。

【0197】

要求がRead a Mailであったときは、まずメールサーバのユーザパスワードを取得し、メールのセッションをオープンする。指定されたメールを読んだ後、セッションをクローズする(ステップS4902~4905)。

【0198】

Print a Mail要求の場合には、Read a Mail要求と同様の処理を行った後、メールをファイルへ書き出し、Jobメッセージを作成した後、リモートデバイスリストの最初のデバイスへJobを送信する(ステップS4906~4912)。

【0199】

どちらの場合も、最後に対応するHTMLページを作成し、作成したページをクライアントコンポーネントに送信して終わる(ステップS4913~4914)。

【0200】

図50は、スケジュールに関する要求の処理手順を示すフローチャートである。クライアントの要求が、「View Schedule」「Print Schedule」であった場合のデスクサーバの処理手順を表す。

【0201】

要求がView Scheduleあった場合、スケジュールのタイプがDailyかMonthlyかを調べ、スケジュールデータを取得する(ステップS5002～5003)。要求がPrint Scheduleの場合は上記と同様の処理を行った後、データを印刷用ファイルに書き出し、Jobメッセージを作成して、リモートデバイスリストの最初のデバイスへJobを送信する(ステップS5004～5009)。

【0202】

どちらの場合も、対応するHTMLページを作成し、そのページをクライアントコンポーネントに送信して処理を終える(ステップS5010～5011)。

【0203】

図51は、他のデバイスに関する要求の処理手順を示すフローチャートである。クライアントの要求が、「Goto Other Device」であった場合のデスクのサーバの処理手順を表す。

【0204】

要求がGoto Other Deviceの場合、デバイス名とデバイスアドレスを取得する。その後、そのデバイスをリモートデバイスリストに追加し,login verbを作成する(ステップS5102～5104)。Login verbによって、デバイスへログインし、対応するHTMLページを作成し、最後にクライアントコンポーネントにページを送信して終わる。この処理は、図16につき説明したものと同様である。

【0205】

図52は、ログアウト及びヘルプに関する要求の処理手順を示すフローチャートである。クライアントの要求が、「logout」「help」であった場合のデスクのサーバの処理手順を表す。

【0206】

Logout要求の場合には、ステップS5202でリモートデバイスリストが空かを調べ、空のときは、ステップS5205で、ステータスに従ってOpeningScreenページを作成する。空でないときは、リストが空になるまで、ステップS5203で次のデバイスにlogin verb=Disconnectでログインし、ステップS5204でそのデバイスをリモートデバイスリストから削除することを繰り返す。すなわち、リモートデバイ

リスト中の全てのデバイスをログアウトする。リストが空になったら、ステップS5205で、OpeningScreenページを作成する。

【0207】

Help要求の場合、ヘルプコンテキストパラメータを取得し、対応するヘルプスクリーンHTMLページを作成する(ステップS5206～5207)。

【0208】

どちらの場合も作成したHTMLページをクライアントコンポーネントに送信して処理を終える。

【0209】

図53は、初期メニュー画面からの処理手順を示すフローチャートである。図106に、初期メニューの画面例を示す。ここで、プリントキューにジョブがある場合には、ステータス画面に一覧表示される。各ジョブの情報としては、ジョブ名、送信者が表示される。

【0210】

初期メニューでは、ステップS5301で、ユーザは自分の希望する項目を選択する。ステップS5302で、選択項目が判別され、その項目に従って新しい画面が表示される。

【0211】

まず、「new interaction (新しい操作)」項目を選択すると、ステップS5303において、図55につき後述する新規の処理(ログイン処理)に進み、画面は「ログイン」画面(図112)へと遷移する。

【0212】

「Print URL (URLの印刷)」項目を選択すると、画面は「URLの印刷」画面(図107)へと遷移する。ステップS5311において、URLを直接入力あるいは参照により指定して「Go」を選択すると、ステップS5312で情報が取得され、プレビューされる。ステップS5313で、確認した内容を印刷する。その後「初期メニュー」画面へ戻る。

【0213】

「Public Info (公開情報)」項目を選択すると、ステップS5314に

おいて、図54につき後述する公開情報の処理に進み、画面は「公開情報の処理」画面（図110）へと遷移する。

【0214】

「Goto Device（他のデバイス）」項目を選択すると、ステップS5315において、他のデバイスにアクセスするための設定画面（図81）が表示される。以後、図51につき説明した他のデバイスへアクセスする処理の「Device」の場合の処理が行われる。

【0215】

また、この初期画面では、プリントキューのジョブを選択することができる。ジョブを選択するには、所望のジョブ名の前に表示されたボタンにタッチする。このようにして「特定のJob」が選択されると、ステップS5304においてパスワード入力画面（図108）が表示される。ここで、パスワードを入力し、選択されたJobにつけられたパスワードと一致すると、プリントキュー内で選択されたジョブに対する処理の設定画面（図109）が表示される。

【0216】

続いてJobに対するActionの選択がステップS5305において行われ、ステップS5306では、アクションに関する時間が設定される。アクションとして「後で印刷（Print Later）」を選択した場合、印刷時期が設定できる。ここで、印刷時期は、具体的な日時（絶対日時）によって設定できるとともに、「1時間後（After 1 Hour）」のように設定時点からの経過時間による相対的な日時でも設定できる。この設定のためには、「1時間後」、「2時間後」、「6時間後」、「明日」、「明後日」、「1週間後」、「2週間後」のような項目をメニューとして用意しておき、その中から選択できるようにする。また、絶対的な日時で設定する場合、日付は現在の日付をデフォルト値とし、他の日付を選ぶ場合に、カレンダをポップアップさせ、該当する日付にタッチして入力する。更に必要であれば時刻を現在時刻から変更する。

【0217】

「保留（Hold Here）」を選択した場合、保留期間を設定する。この場合も、後で印刷する場合と同様にして、保留期限を絶対日時で指定できるとと

もに、保留する期間、例えば1時間（For 1 Hour）のような設定をメニュー選択できるようにする。また、以上の絶対日時、相対日時の指定の場合に、タッチキーボードを表示させて、直接数値を入力してもよいし、特定のボタンの指示回数に応じて、表示値をカウントアップ・ダウントするようにしてもよい。

【0218】

以上のように選択・設定されたアクションや日時の項目に従って、Actionが設定・実行される。まず、「後で印刷」が選択されると、ステップS5308で、印刷ジョブがキューから削除されて、印刷時刻とともに保留ジョブのリストに登録され、「初期メニュー」画面へ戻る。その後、設定時刻になると、印刷が行われる。

【0219】

また、「保留」が選択されると、ステップS5309で、印刷ジョブがキューから削除されて、保留期限とともにアクションとして保留が設定され、保留ジョブのリストに登録され、「初期メニュー」画面へ戻る。その後、保留期間の間、Jobは保留され、期限までに何もしないと、取り消される。

【0220】

また、「ポーズ (Pause/Restart Printing)」が選択されると、ステップS5310で、印刷を一時停止し、「初期メニュー」画面へ戻り、後で「再スタート (Pause/Restart Printing)」が選択される（同じボタンを再押下する）のを待つ。「印刷キャンセル (Cancel Printing)」が選択されると、印刷ジョブがキューから削除され、履歴にこのジョブのアクションとして「キャンセル」が記録され、「初期メニュー」画面へ戻る。

【0221】

図54は、公開情報の処理手順を示すフローチャートである。初期メニューにおいて「公開情報の処理」項目が選択された場合の画面遷移と処理の流れを示したものである。まず、ステップS5401で、このプリンタ上の公開情報のリストを表示する。図110は、そのような公開情報のリストとともに公開情報の処理設定画面を表示した例を示す図である。このリストに表示されるのは、公開情報と

いう属性を持つ情報（公開というアクションを持つジョブ）であり、パスワード無しで印刷することができる。

【0222】

ステップS5402で、リストから特定のジョブを選択した場合には、ステップS5405に進み、選択されたジョブに対する処理を行なう。「他のデバイスから（From Other Device）」を選択した場合、他のデバイスにある公開情報を処理するために、ステップS5415に進む。

【0223】

ステップS5405では、選択されたジョブに対する処理を選択する。「印刷」を選択した場合には、ステップS5407で、通常のジョブの印刷処理につき後で詳述する印刷場所や必要なら印刷時刻の設定を実行し、ステップS5408で選択された公開情報の印刷を実行する。「プレビュー」を選択した場合には、ステップS5409でプレビューを実行し、ステップS5410で次の処理を選択する。

【0224】

ステップS5406或いはS5410で「追加（Make Job As Public Info. On Other Device）」を選択した場合には、選択した情報を他のデバイス上で公開するために、公開するためのデバイスを指定して、選択した公開情報をそのデバイスの公開リストに追加する設定を行なう。図111は、情報を他のデバイス上で公開するための設定画面の例を示す図である。

【0225】

ステップS5412では、追加先として、デバイスのアドレスと装置名を設定し、更に、いつまで公開するか、すなわち、公開期限を設定する。この場合も、前述のように、公開期限を絶対日時で設定してもよいし、現時点からの公開期間を設定してもよい。また、すぐに公開したくなれば、公開を開始する日時を設定してもよい。この場合も、「1週間後」のような相対的な設定も可能である。公開期間を制限するには、公開情報のリストを表示する際に、設定された期間をチェックして期間外のものを対象外とすればよい。また、定期的にリストをチェックして、時間が過ぎたものをリストから削除するようにしてもよい。

【0226】

一方、ステップS5415では、他のデバイスにアクセスするために、既に述べたように、他のデバイスにアクセスするための設定画面（図81）を表示し、デバイスのアドレスと名前を指定してアクセスする。ステップS5416では、アクセスした装置から公開情報リストを取得して、表示する。以後は、自装置における装置と同様に処理が実行される。

【0227】

図55は、ログイン処理手順を示すフローチャートである。初期メニューにおいて新規の処理を行うと選択された場合の処理である。まず、ステップS5501で、「ログイン」画面（図112）を表示し、入力されたユーザの識別子をチェックし、ログインが許可されれば、ステップS5502で、今ログインしたユーザに有効なジョブ（処理が実行されていない保留されたジョブ）があるかをチェックする。ジョブがあれば、ステップS5504で、そのジョブのリスト（図113）を表示して、図56につき後述するジョブを選択してアクションを行なう処理に進む。ジョブがなければ、ステップS5503で、メインメニュー（図121）を表示して、図64につき後述するメインメニューからの処理へ進む。

【0228】

図56は、ジョブを選択してアクションを行なう処理手順を示すフローチャートである。ユーザのログイン時にそのユーザに対して有効なJobが存在する場合の処理である。ステップS5601で、ユーザの保留されたジョブのリストを表示する。図113は、保留されたジョブのリストの例を示す図である。

【0229】

ここで、保留されているジョブには、設定された日時に印刷が予定されているジョブ、設定された日時に送信が予定されているジョブ、設定された日時まで単に保持されている（その時が来るまでに何もしないと取り消される）ジョブがある。リストには、ジョブ名、送信者あるいは受信者、アクションの種類（印刷、送信、保持）、アクションの実行予定日時などが、表示される。これらのジョブは、アクションの種類に応じて、色を変えたり、異なるマークを付けるなど、種類が識別可能な表示態様で表示される。なお、装置には、実行済みのジョブや、

実行が取り消されたジョブも、ステータスを「実行済み」や「取り消し」に変えることで、履歴として記録されており、これらも必要に応じて、実行前のジョブとともにリストに表示させることができる。その場合、ステータスによっても、表示態様を変えるものとする。

【0230】

次に、ステップS5602で対象となるジョブを選択し、ステップS5603で、選択されたジョブに対する処理（アクション）を選択する。ステップS5603で、メインメニューが選択された場合には、ステップS5605に進み、図64につき後述するメインメニューからの処理を行なう。検索が選択された場合には、ステップS5606に進み、図65につき後述する検索処理を行なう。印刷が選択された場合には、ステップS5607に進み、図60につき後述する選択Jobの印刷処理を行なう。送信が選択された場合には、ステップS5610に進み、図62につき後述する選択Jobの送信処理を行なう。プレビューが選択された場合には、ステップS5608に進み、図57につき後述する選択Jobのプレビュー処理を行なう。削除が選択された場合には、ステップS5609に進み、図58につき後述する選択Jobの削除処理を行なう。スケジュールが選択された場合には、ステップS5611に進み、図59につき後述する選択Jobのスケジュール変更処理を行なう。ログアウトが選択された場合には、ステップS5612に進み、図52につき説明したログアウト処理を行なう。

【0231】

図57は、選択Jobのプレビュー処理手順を示すフローチャートである。有効なJobのリストから選択したJobに対して、プレビューを実行する際の処理の流れを表したものである。ステップS5701で、選択されたジョブのプレビューを実行する。図114は、ジョブのプレビューの画面表示例を示す図である。ステップS5702で、選択されたジョブに対する処理を選択する。

【0232】

ステップS5702で、印刷が選択された場合には、ステップS5704に進み、図60につき後述する選択Jobの印刷処理を行なう。送信が選択された場合には、ステップS5706に進み、図62につき後述する選択Jobの送信処理を行なう。削除が選択された場合には、ステップS5705に進み、図58につき後述する選択Jobの削除処理を行なう。

行なう。スケジュールが選択された場合には、ステップS5707に進み、図59につき後述する選択Jobのスケジュール変更処理を行なう。キャンセルが選択された場合には、ステップS5708に進み、図56につき説明したJobのリスト表示に戻る。

【0233】

図58は、選択Jobの削除処理手順を示すフローチャートである。有効なJobのリストから選択したJobに対して削除を行う場合の処理の流れを示したものである。まず、ステップS58001で、削除されるジョブの識別情報とともにジョブのプレビューを行ない、ユーザに削除の確認を促す。図115は、削除の確認画面を示す図である。ここで、削除が選択されれば、ステップS5802で、ジョブの削除を実行する。この時、図97につき説明した処理が実行される。ここでは、選択されたジョブは、保留されているジョブのリストから選択されているので、アクションステータスは未実行であり、削除されたジョブはキャンセルというステータスとなってユーザの履歴として記録される。続いてステップS5803に進み、図56につき説明したJobのリスト表示に戻る。

【0234】

図59は、選択Jobのスケジュール変更処理手順を示すフローチャートである。有効なJobのリストから選択したJobに対してスケジュールの変更を行う場合の処理の流れを示したものである。ステップS5901で、変更対象となる現在のスケジュールを表示する。図116は、スケジュールの変更のための設定画面を示す図である。ステップS5902で、スケジュールの変更を行なう。続いてステップS5903に進み、図56につき説明したJobのリスト表示に戻る。

【0235】

図60は、選択Jobの印刷処理手順を示すフローチャートである。有効なJobのリストから選択したJobに対して印刷を行う場合の処理の流れを示したものである。ステップS6001で、選択ジョブの元々のアクションのタイプが判別される。元々のアクションが印刷である場合は、図61につき後述する印刷設定変更処理が実行される。元々のアクションが印刷でない場合は、ステップS6002で、印刷設定メニューが表示される。図117は、印刷設定メニューの例を示す図である。設定項目としては、どこで印刷するか（この装置か別の装置か）、別の装置の場合に

は、その装置名とアドレス、いつ印刷するかなどがある。

【0236】

別の装置で印刷する場合には、ジョブの履歴におけるアクションの種類として、送信が記録される。また、別の装置で印刷する際に、送信処理につき後述するセンド方式で送信してもよいし、トランスマスター方式で送信するようにしてよい。

【0237】

ステップS6003で、処理を選択する。印刷を選択した場合は、ステップS6005で、ユーザの設定内容を装置に設定し、すぐに印刷する場合は印刷を実行する。続いてステップS6006に進み、図56につき説明したJobのリスト表示に戻る。プレビューを選択した場合は、ステップS6007でプレビューを実行し、その後、印刷が必要であれば、ステップS6005に進む。通知を選択した場合は、ステップS6009で通知の設定を行い、印刷設定メニューに戻る。図118は、通知設定画面の例を示す図である。通知の設定項目としては、通知相手の人物名、通知媒体（ボイスメッセージ、ポップアップウィンドウ、電子メール、電話、FAX、ポケットベル）、通知先（アドレス、電話番号など）、通知の契機（成功時、失敗時、アクセス時）、通知の契機から実際に通知するまでの時間などがある。また、この画面では、各通知について設定内容を含んだ通知のリスト表示ができる。

【0238】

図61は、選択Jobの印刷設定の変更処理手順を示すフローチャートである。Jobに対する選択されたJobが元々のActionタイプが印刷である場合の状態遷移と処理の流れを示したものである。

【0239】

ステップS6101で、印刷設定変更メニューが表示される。図119は、印刷設定変更メニューの例を示す図である。ステップS6102で、ジョブに対する処理を選択する。「現時点でも印刷（Print Now Also）」を選択した場合、ステップS6104で、選択ジョブとは別に、選択ジョブの印刷時期を現時点としたジョブを追加する。「現時点で印刷（Print Now）」を選択した場合、選択ジョブの印刷時期を現時点に変更する。従って、これらの場合、現時点で印刷

が行われる。「スケジュール」を選択した場合、ステップS6106で、図116のスケジュール変更メニューを表示する。ステップS6107で、メニューを使用して印刷スケジュールを変更する。いずれの場合も、続いてステップS6108に進み、図56につき説明したJobのリスト表示に戻る。

【0240】

図62は、選択Jobの送信処理手順を示すフローチャートである。有効なJobリストから選択したJobに対して送信を行う場合の状態遷移と処理の流れを示したものである。ステップS6201で、選択ジョブの元々のアクションのタイプが判別される。元々のアクションが送信である場合は、図63につき後述する送信設定変更処理が実行される。元々のアクションが印刷でない場合は、ステップS6202で、送信設定メニューが表示される。図70は、送信設定メニューの例を示す図である。

【0241】

ステップS6203で、設定を行ない、処理を選択する。送信先、送信時期について設定を行なう。これらの設定は、印刷の場合と同様である。処理として送信（Send）を選択した場合には、ステップS6205で、選択されたジョブがコピーされ、送信される。この場合、選択されたジョブはジョブのリストに残る。一方、処理として転送（Transfer）を選択した場合には、ステップS6206で、選択されたジョブが送信され、選択されたジョブはジョブのリストから削除される。送信、転送のいずれの場合も、続いてステップS6209に進み、図56につき説明したJobのリスト表示に戻る。また、通知を選択した場合には、ステップS6207で、通知の設定画面を表示して、設定を行なう。設定後に、送信、転送の選択のために送信設定メニューの表示に戻る。

【0242】

図63は、選択Jobの送信設定の変更処理手順を示すフローチャートである。これは、選択されたJobの元々のActionタイプが送信である場合の状態遷移と処理の流れを示したものである。

【0243】

ステップS6301で、送信設定変更メニューが表示される。図71は、送信設定変

更メニューの例を示す図である。ステップS6302で、ジョブに対する処理を選択する。「現時点でも送信（Send Now Also）」を選択した場合、ステップS6304で、選択ジョブとは別に、選択ジョブの送信時期を現時点としたジョブを追加する。「現時点で送信（Send Now）」を選択した場合、選択ジョブの送信時期を現時点に変更する。従って、これらの場合、現時点で送信が行われる。「スケジュール」を選択した場合、ステップS6106で、図116のスケジュール変更メニューを表示する。ステップS6307で、メニューを使用して送信スケジュールを変更する。いずれの場合も、続いてステップS6308に進み、図56につき説明したJobのリスト表示に戻る。また、「修正（Modify）」が選択された場合、ステップS6309で、修正メニューを表示する。

【0244】

図120は、修正メニューの例を示す図である。このとき、送信先が一覧表示される。ステップS6310では、新たな送信先の追加・設定、一覧より選択した送信先の修正（削除、変更）、送信先で実行すべき処理の変更、パスワードの変更、送信時期の変更などを行なう。通知の設定を行なう場合は、ステップS6312で通知設定メニューを表示して設定を行ない、修正メニューに戻る。通知の設定を行わない場合は、送信設定変更メニューに戻る。

【0245】

図64は、メインメニューからの処理手順を示すフローチャートである。これは、メインメニューでの画面遷移と処理の流れを示したものである。ステップS6401で、メインメニューを表示する。図121は、メインメニューの例を示す図である。ステップS6402で、処理（アクション）を選択する。

【0246】

ステップS6402で、ログアウトが選択された場合には、ステップS6404に進み、図52につき説明したログアウト処理を行なう。ヘルプが選択された場合には、ステップS6405に進み、図52につき説明したヘルプ処理を行なう。検索が選択された場合には、ステップS6406に進み、図65につき後述する検索処理を行なう。印刷が選択された場合には、ステップS5607に進み、図67につき後述する印刷処理を行なう。送信が選択された場合には、ステップS6408に進み、図69につき後述

する送信処理を行なう。削除が選択された場合には、ステップS6409に進み、図75につき後述する削除処理を行なう。命令の設定が選択された場合には、ステップS6410に進み、図77につき後述する命令の設定処理を行なう。スケジュールが選択された場合には、ステップS6411に進み、図78につき後述するスケジュール変更処理を行なう。公開情報の追加が選択された場合には、ステップS6412に進み、図66につき後述する公開情報の追加処理を行なう。他のデバイスが選択された場合には、ステップS6413に進み、図80につき後述する他のデバイスへのアクセス処理を行なう。デスクトップが選択された場合には、ステップS6414に進み、図80につき後述するデスクトップへのアクセス処理を行なう。

【0247】

図65は、検索処理手順を示すフローチャートである。これは、メインメニューから「検索」が選択された場合の処理の流れを示したものである。

【0248】

ステップS6501で、ジョブを検索するための検索条件を設定する。検索条件は、ジョブの設定時期、実行時期、実行状態、送信者などが利用でき、検索は、保留中のジョブに加えて、実行済みの履歴中のジョブも対象になる。特に、実行状態が実行済み（必要であれば、キャンセルも）であることを条件として検索すれば、履歴を表示させることができる。ステップS6502で検索を実行し、検索結果のリストを表示させる。ステップS6503で、検索結果よりジョブを選択し、ステップS6504で、選択したジョブに対する処理を選択する。

【0249】

ステップS6504で、印刷が選択された場合には、ステップS6506に進み、図60につき説明した選択Jobの印刷処理を行なう。送信が選択された場合には、ステップS6509に進み、図62につき説明した選択Jobの送信処理を行なう。プレビューが選択された場合には、ステップS6507に進み、図57につき説明した選択Jobのプレビュー処理を行なう。削除が選択された場合には、ステップS6508に進み、図58につき説明した選択Jobの削除処理を行なう。スケジュールが選択された場合には、ステップS6510に進み、図59につき説明した選択Jobのスケジュール変更処理を行なう。メインメニューが選択された場合には、ステップS6511に進み、図64

につき説明したメインメニューからの処理を行なう。

【0250】

図66は公開情報への追加処理手順を示すフローチャートである。これは、「公開情報への追加」を選択した際の処理の流れを示したものである。ステップS6601で、公開情報とするジョブをジョブのリストから選択する。ステップS6602で、公開情報への追加メニュー（図111）を表示して、追加先、公開期限の設定を行なう。この詳細は、公開情報の処理で説明したものと同様である。但し、ここでは、公開する対象は、ユーザのジョブリスト中のジョブである。設定後、ステップS6603で、追加を指示すると、ステップS6605で、選択したジョブが設定した公開情報のリストに追加される。処理の終了後、キャンセルを選択すると、メインメニューへ戻る。

【0251】

図67は、印刷処理手順を示すフローチャートである。これは、メインメニューから「印刷」を選択した際の処理の流れを示したものである。

【0252】

ステップS6701で、印刷対象のジョブのある場所を選択する。図122は、印刷対象のジョブのある場所を選択するためのメニュー画面の例を示す図である。現在のデバイスを選択した場合、ステップS6703で、現在のデバイスの印刷Jobのリストを表示する。図123は、印刷対象のジョブを選択するためのメニューの例を示す図である。ステップS6704で、印刷するジョブを選択する。ステップS6705で、選択ジョブの印刷の設定を行なう。これは、図60につき説明したものと同様である。

【0253】

ジョブのある場所として、他のデバイスまたはデスクトップを選択した場合、ステップS6706で、デバイスのアドレスを指定し、ステップS6707で、そのアドレスのデバイスからログイン画面をダウンロードする。ステップS6708で、この画面にしたがって、ログインを実行する。ログインができたなら、ログインver bが印刷であるので、図51につき説明したように、ステップS6710で、図122の印刷対象のジョブを選択するためのメニュー画面がダウンロードされ、ここで

、印刷デバイスを選択する。現在のプリンタで印刷する場合、ステップS6711に進み、図68につき説明するプリンタによるリモートプリント処理を行なう。デスクトップで印刷する場合、ステップS6712に進み、デスクトップによるリモートプリント処理を行なう。その他のデバイスで印刷する場合、ステップS6713に進み、他のデバイスによるリモートプリント処理を行なう。

【0254】

図68は、プリンタによるリモートプリント処理手順を示すフローチャートである。メインメニューから他のデバイスでの印刷を指定して、そのデバイスが同種のプリンタであった場合の、ログイン後における処理の流れを示している。処理の手順は、現在のデバイス上のジョブの印刷と同様である。但し、メニューやりストは、アクセス先の装置からダウンロードされる。

【0255】

図69は、送信処理手順を示すフローチャートである。これは、メインメニューから「送信」を選択した際の画面遷移と処理の流れを示したものである。

【0256】

ステップS6901で、送信対象ジョブのある場所は、このデバイスか、他のデバイスまたはデスクトップかを選択する。図124は、送信対象のジョブのある場所を選択するためのメニュー画面の例を示す図である。このデバイスを選択した場合、ステップS6902で、デバイス上のジョブのリストを表示する。図125は、送信対象のジョブを選択するためのメニューの例を示す図である。ステップS6903で、送信対象とするジョブを選択する。以後は、図62につき説明した選択ジョブの送信処理を行なう。

【0257】

送信対象ジョブのある場所として、他のデバイスまたはデスクトップを選択した場合、ステップS6904で、デバイスのアドレスを指定し、ステップS6905で、そのアドレスのデバイスからログイン画面をダウンロードする。ステップS6906で、この画面にしたがって、ログインを実行する。ログインができたなら、ログインverbが送信であるので、図51につき説明したように、ステップS6908で、図124の送信対象のジョブを選択するためのメニュー画面がダウンロードされ

、ここで、送信デバイスを選択する。プリンタから送信する場合、ステップS6909に進み、図72につき説明するプリンタによるリモートセンド処理を行なう。デスクトップから送信する場合、ステップS6910に進み、デスクトップによるリモートセンド処理を行なう。その他のデバイスから送信する場合、ステップS6911に進み、他のデバイスからのリモートセンド処理を行なう。

【0258】

図72は、プリンタによるリモートセンド処理手順を示すフローチャートである。メインメニューから他のデバイスからの送信を指定して、そのデバイスが同種のプリンタであった場合の、ログイン後における処理の流れを示している。

【0259】

そのデバイス上にジョブが存在する場合に、ステップS7202で、送信ジョブのリストを表示する。ステップS7203で、ジョブのリストから送信するジョブを選択する。ステップS7204で、処理を選択する。ここで、送信処理を選択した場合は、選択ジョブの送信処理が行われる。この処理の手順は、図69につき説明した現在のデバイス上の選択ジョブの送信と同様である。但し、メニューやリストは、アクセス先の装置からダウンロードされる。

【0260】

ステップS7204で、検索を選択した場合には、ステップS7206で、送信対象ジョブを新たに検索する。ログアウトが選択された場合には、ステップS7207に進み、図52につき説明したログアウト処理を行なう。メインメニューが選択された場合には、ステップS7208で、メインメニューの表示に戻る。

【0261】

図75は、削除処理手順を示すフローチャートである。これは、メインメニューから削除を選んだ際の処理の流れを示したものである。

【0262】

ステップS7501で、削除対象ジョブのある場所は、このデバイスか、他のデバイスまたはデスクトップかを選択する。図73は、削除したいJobの存在する装置を選択するためのメニューの例を示す図である。このデバイスを選択した場合、ステップS7502で、デバイス上のジョブのリストを表示する。図74は、削除Job

のリストの例を示す図である。ステップS7503で、送信対象とするジョブを選択する。以後は、図58につき説明した選択ジョブの削除処理を行なう。

【0263】

削除対象ジョブのある場所として、他のデバイスまたはデスクトップを選択した場合、ステップS7505で、デバイスのアドレスを指定し、ステップS7506で、そのアドレスのデバイスからログイン画面をダウンロードする。ステップS7507で、この画面にしたがって、ログインを実行する。ログインができたなら、ログインverbが削除であるので、図51につき説明したように、ステップS7509で、図74の削除対象のジョブを選択するためのメニュー画面がダウンロードされ、ここで、ジョブを選択する。プリンタから削除する場合、ステップS7510に進み、図76につき説明するプリンタによるリモートデリート処理を行なう。デスクトップから削除する場合、ステップS7511に進み、デスクトップによるリモートデリート処理を行なう。その他のデバイスから削除する場合、ステップS7512に進み、他のデバイスからのリモートデリート処理を行なう。

【0264】

図76は、プリンタによるリモートデリート処理手順を示すフローチャートである。メインメニューから他のデバイスからの送信を指定して、そのデバイスが同種のプリンタであった場合の、ログイン後における処理の流れを示している。

【0265】

そのデバイス上にジョブが存在する場合に、ステップS7601で、ジョブのリストを表示する。ステップS7602で、ジョブのリストから削除するジョブを選択する。ステップS7603で、処理を選択する。ここで、削除処理を選択した場合は、ステップS7605で、図115の削除確認メニューが表示され、ステップS7606で、選択ジョブの削除処理が行われる。この処理の手順は、図58につき説明した現在のデバイス上の選択ジョブの削除と同様である。但し、メニュー・リストは、アクセス先の装置からダウンロードされる。

【0266】

ステップS7604で、検索を選択した場合には、ステップS7607で、削除対象ジョブを新たに検索する。ログアウトが選択された場合には、ステップS7608に進

み、図52につき説明したログアウト処理を行なう。メインメニューが選択された場合には、ステップS7609で、メインメニューの表示に戻る。

【0267】

図77は、命令の設定処理手順を示すフローチャートである。これは、メインメニューから命令の設定を選択した際の、画面遷移と処理の流れを示したものである。ステップS7701で、命令設定画面を表示する。図126は、命令設定画面の例を示す図である。ここでは、命令の実行条件を設定する。条件としては、ジョブ名、デバイス名、ユーザ名、命令の実行の契機となる状況（ジョブの受信時、送信中、印刷時、ステータスの変更など）、命令の有効期限あるいは期間（無期限を含む）、プリンタの状態が設けられており、これらのANDかORなども設定できる。図126に示す「On receiving jobs from others」のように、現在の設定内容が表示される。

【0268】

ステップS7702で、処理が選択される。ここで、「アクションの設定（Task Action）」を選択すると、ステップS7704へ進み、設定した条件が満たされたときに行なうべきアクションの設定を行なう。ここでは、ジョブの印刷、保留が設定できる。更に、ジョブの送信（転送）、キャンセルなどが設定できてもよい。通知を選択すると、ステップS7705で、通知の設定を行なう。命令の設定を選択すると、画面上で設定した命令が有効となる。命令の削除を選択すると、画面上の設定対象の命令が削除される。その後、ステップS7707で、メインメニューに戻る。有効期限を過ぎた命令は、実行済みのジョブと同様に履歴に残り、履歴の管理において、期限より更に一定期間が過ぎると削除される。

【0269】

図78は、スケジュール変更処理手順を示すフローチャートである。これは、メインメニューからスケジュールの変更を選択した際の処理の流れを示したものである。

【0270】

ステップS7801で、スケジュールの変更画面を表示して、変更対象ジョブのある場所は、このデバイスか、他のデバイスまたはデスクトップかを選択する。図

127は、変更したいJobの存在する装置を選択するためのメニューの例を示す図である。現在のデバイスを選択した場合、ステップS7803で、デバイス上のジョブのリストを表示する。図128は、スケジュールの変更対象Jobの選択画面の例を示す図である。ステップS7804で、変更対象とするジョブを選択する。ステップS7805で、図59につき説明した選択ジョブのスケジュールの変更を行なう。

【0271】

変更対象ジョブのある場所として、他のデバイスまたはデスクトップを選択した場合、ステップS7806で、デバイスのアドレスを指定し、ステップS7807で、そのアドレスのデバイスからログイン画面をダウンロードする。ステップS7808で、この画面にしたがって、ログインを実行する。ログインができたなら、ログインverbがスケジュールの変更であるので、図51につき説明したように、ステップS7810で、図128のスケジュール変更対象のジョブを選択するためのメニュー画面がダウンロードされ、ここで、ジョブを選択する。プリンタのスケジュールを変更する場合、ステップS7811に進み、図79につき説明するプリンタによるリモートリストスケジュール処理を行なう。デスクトップのスケジュールを変更する場合、ステップS7812に進み、デスクトップによるリモートリストスケジュール処理を行なう。

【0272】

図79は、リモートリストスケジュール処理処理手順を示すフローチャートである。メインメニューから他のデバイス（同種のプリンタ）のJobのスケジュール変更を選択した場合の処理の流れを示している。

【0273】

デバイス上にジョブが存在する場合に、ステップS7902で、ジョブの全リストを表示する。ステップS7903で、処理を選択する。ここで、ジョブを選択し、スケジュールの変更処理を選択した場合は、ステップS7905で、図116の選択ジョブのスケジュールの変更メニューが表示され、選択ジョブのスケジュールの変更処理が行われる。この処理の手順は、図59につき説明した現在のデバイス上の選択ジョブのスケジュールの変更と同様である。但し、メニュー・リストは、アクセス先の装置からダウンロードされる。

【0274】

ステップS7904で、検索を選択した場合には、ステップS7906で、スケジュールの変更対象ジョブを新たに検索する。ログアウトが選択された場合には、ステップS7907に進み、図52につき説明したログアウト処理を行なう。メインメニューが選択された場合には、ステップS7908で、メインメニューの表示に戻る。

【0275】

図80は、他のデバイスへのアクセス処理手順を示すフローチャートである。これは、メインメニューから他のデバイスへのアクセス、またはデスクトップへのアクセスを選択した場合の処理の流れを示している。ステップS8001で、デバイスのアドレスを指定し、ステップS8002で、そのアドレスのデバイスからログイン画面をダウンロードする。ステップS8003で、この画面にしたがって、ログインを実行する。ログインができたなら、ログインverbがデバイスであるので、図51につき説明したように、ステップS8005で、図106の初期メニュー画面がダウンロードされ、操作対象デバイスを判別する。操作対象がプリンタである場合、ステップS8006に進み、自プリンタと同様に、他のプリンタの処理を行なう。操作対象がデスクトップである場合、ステップS8007に進み、図82につき後述するデスクトップの処理を行なう。操作対象が他のデバイスである場合、ステップS8008に進み、他のデバイスの処理を行なう。

【0276】

図82は、デスクトップの処理手順を示すフローチャートである。メインメニューからGoto My Desktopで自分のデスクを選んだ時の初期画面の画面遷移と処理の流れとを示している。まず、ステップS8201で、メインメニューを表示する。図129は、デスクトップメインメニューの例を示す図である。ステップS8202で、処理（アクション）を選択する。

【0277】

ステップS8202で、ファイル印刷が選択された場合には、ステップS8204に進み、図88につき後述するファイル印刷処理を行なう。未読メール取得が選択された場合には、ステップS8205に進み、図83につき後述する未読メール取得処理を行なう。既読メール取得が選択された場合には、ステップS8206に進み、図84につ

き後述する既読メール取得処理を行なう。スケジュール印刷が選択された場合には、ステップS8207に進み、スケジュール印刷処理を行なう。スケジュール参照が選択された場合には、ステップS8208に進み、スケジュール参照処理を行なう。アドレス帳参照が選択された場合には、ステップS8209に進み、アドレス帳参照処理を行なう。ログアウトが選択された場合には、ステップS8210に進み、図5-2につき説明したログアウト処理を行なう。

【0278】

図83は、未読メール処理手順を示すフローチャートである。これは、自分のデスクにアクセスして、未読メールを読もうとした際の処理の流れを示している。ステップS8300で、未読メールのリストを表示する。図85は、未読メールのリストの例を示す図である。ステップS8301で、処理対象のメールを選択する。ステップS8302で、この選択されたメールに対する処理を選択する。

【0279】

ステップS8302で、メインメニューが選択された場合には、ステップS8304で、デスクトップのメインメニューに戻る。「開く」が選択された場合には、ステップS8305で、選択されたメールを開いて内容を表示する。図86は、メールの内容の表示例を示す図である。ステップS8306で、前の表示や後の表示が選択された場合には、ステップS8305に戻り、対応するメールの内容を表示する。印刷や削除が選択された場合には、それぞれ、ステップS8308のメールの削除、ステップS8309のメールの印刷が実行される。ステップS8302で、印刷や削除が選択された場合にも、それぞれ、ステップS8308のメールの削除、ステップS8309のメールの印刷が実行される。全て印刷が選択された場合には、ステップS8310で、全てのメールが印刷される。「既読メール（Old）」が選択された場合には、図84につき後述する既読メールの処理を行なう。

【0280】

メール印刷は自分のデスクにアクセスして、メールを取得して、印刷する。印刷に関しては、前述のジョブの印刷と同様に印刷先などを指定できる。全メール印刷は自分のデスクにアクセスして、全てのメールを取得して、印刷する。メール削除は、自分のデスクにアクセスして、メールを削除する。その際、ジョブの

削除と同様に削除の確認が行なわれるようにしてよい。

【0281】

図84は、既読メール処理手順を示すフローチャートである。これは、自分のデスクにアクセスして、既読メールを読もうとした際の処理の流れを示している。ステップS8401で、未読メールのリストを表示する。図87は、既読メールのリストの例を示す図である。ステップS8402で、処理対象のメールを選択する。ステップS8403で、この選択されたメールに対する処理を選択する。

【0282】

ステップS8404で、メインメニューが選択された場合には、ステップS8405で、デスクトップのメインメニューに戻る。「開く」が選択された場合には、ステップS8406で、選択されたメールを開いて内容を表示する。図86は、メールの内容の表示例を示す図である。ステップS8407で、前の表示や後の表示が選択された場合には、ステップS8406に戻り、対応するメールの内容を表示する。印刷や削除が選択された場合には、それぞれ、ステップS8409のメールの削除、ステップS8410のメールの印刷が実行される。ステップS8302で、印刷や削除が選択された場合にも、それぞれ、ステップS8409のメールの削除、ステップS8410のメールの印刷が実行される。「未読メール（Unread）」が選択された場合には、図83につき説明した未読メールの処理を行なう。

【0283】

図88は、ファイル処理の処理手順を示すフローチャートである。これは、自分のデスクにアクセスして、デスクトップ上のファイルに対して操作しようとした際の処理の流れを示している。ステップS8801で、ファイル取得のための設定を行なう。図130は、ファイル取得のための設定画面の例を示す図である。ここでは、ファイルネームが設定できる。ステップS8802で、取得のために実行すべき処理を選択する。

【0284】

ステップS8802で検索を選択した場合、ステップS8803で、検索条件を設定し、ステップS8804で、デスクトップのファイルの検索を実行し、ステップS8805で、設定条件に該当するファイルのリストが表示される。ステップS8802で、ファイ

ル一覧（Show All）を選択した場合、デスクトップのファイルの一覧が表示される。図131は、デスクトップのファイルの一覧表示の例を示す図である。ステップS8807では、検索結果あるいは一覧表示から、ファイルを選択する。このようにして選択した場合、あるいはステップS8802で、ファイルネームを入力して取得を指示した場合、ステップS8808で、該当ファイルがデスクトップから取得される。ステップS8809で、このファイルに対する処理を選択する。図132は、取得ファイルに対する処理の選択画面の例を示す図である。ここで、印刷が選択された場合、ステップS8811で取得したファイルを印刷する。送信が選択された場合、ステップS8812で、取得したファイルを送信する。

【0285】

尚、本発明は、単一の機器からなる装置に適用しても、複数の機器から構成されるシステムに適用してもよい。また、上述した実施形態の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体を、装置あるいはシステムに供給し、装置あるいはシステム内のコンピュータが記憶媒体に格納されたプログラムコードを読み出して実行することによって達成してもよい。

【0286】

更に、装置あるいはシステム内のコンピュータが記憶媒体に格納されたプログラムコードを読み出して実行することによって、上述した実施形態の機能を直接実現するばかりでなく、そのプログラムコードの指示に基づいて、コンピュータ上で稼動しているOSなどの処理により、上述の機能を実現される場合も含まれる。

【0287】

これらの場合、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0288】

【発明の効果】

以上説明したように、本発明によれば、処理対象情報に関する種々の条件が満たされたとき、所望の処理が行なえるという効果がある。

【図面の簡単な説明】

【図 1】

実施形態の機能構成を示した図である。

【図 2】

本実施形態の機能構成の詳細を表した図である。

【図 3】

ユーザの操作に対応した情報の流れを説明した図である。

【図 4】

ユーザの入力をコマンドとして処理する流れを示した図である。

【図 5】

処理結果やステータス変化をユーザに表示する処理の流れを示した図である。

【図 6】

複数のデバイス間における、機能の関係を表した図である。

【図 7】

本実施形態に係るシステム構成図である。

【図 8】

本実施形態に係るプリンタの機能ブロック図である。

【図 9】

電源投入時の起動順序を示すフローチャートである。

【図 10】

クライアントコンポーネントの動作を表すフローチャートである。

【図 11】

クライアントの要求に対するサーバコンポーネントの処理手順を示すフローチャートである。

【図 12】

CancelAllPrinter jobsの処理手順を示すフローチャートである。

【図 13】

GetAllPrinter jobsの処理手順を示すフローチャートである。

【図14】

SetPrinterJobStatusの処理手順を示すフローチャートである。

【図15】

要求「login」「logout」の処理手順を示すフローチャートである。

【図16】

要求「Goto Other Device」「Goto Desktop」の処理手順を示すフローチャートである。

【図17】

要求「Search」「Help」「Receive Job / Receive Notification」の処理手順を示すフローチャートである。

【図18】

IdAnalyzeの処理手順を示すフローチャートである。

【図19】

DUMMYの処理手順を示すフローチャートである。

【図20】

IdAnalyzeNotifyの処理手順を示すフローチャートである。

【図21】

RecieveHttpNotificationの処理手順を示すフローチャートである。

【図22】

要求「Print」「Send」「Delete」「Reschedule」「Preview」の処理手順を示すフローチャートである。

【図23】

PrintJobの処理手順を示すフローチャートである。

【図24】

SendJobの処理手順を示すフローチャートである。

【図25】

SendJobによる送信の実行手順を示すフローチャートである。

【図26】

CancelJobの処理手順を示すフローチャートである。

【図27】

SendNotificationの処理手順を示すフローチャートである。

【図28】

サーバに対する入力情報と処理との対応関係を示す図である。

【図29】

デーモンの起動からの処理手順を示すフローチャートである。

【図30】

デーモンモジュールのNotification処理手順を示すフローチャートである。

【図31】

デーモンが扱うjobの種類と対応する処理との関係を表した図である。

【図32】

PrintPrinterJobの処理手順を示すフローチャートである。

【図33】

Sendjobの処理手順を示すフローチャートである。

【図34】

ReceivejobDataの処理手順を示すフローチャートである。

【図35】

GetListOfJobAttributeの処理手順を示すフローチャートである。

【図36】

PollJobの処理手順を示すフローチャートである。

【図37】

PollNotifyの処理手順を示すフローチャートである。

【図38】

CancelActionの処理手順を示すフローチャートである。

【図39】

EditActionの処理手順を示すフローチャートである。

【図40】

EditActionの処理手順を示すフローチャートである。

【図41】

RescheduleActionの処理手順を示すフローチャートである。

【図42】

RescheduleActionの処理手順を示すフローチャートである。

【図43】

RescheduleActionの処理手順を示すフローチャートである。

【図44】

AddNotificationの処理手順を示すフローチャートである。

【図45】

AddInstructionの処理手順を示すフローチャートである。

【図46】

GetInstructionの処理手順を示すフローチャートである。

【図47】

ExecutionInstructionの処理手順を示すフローチャートである。

【図48】

デスクトップのサーバコンポーネントの処理手順を示すフローチャートである

【図49】

メールに関する要求の処理手順を示すフローチャートである。

【図50】

スケジュールに関する要求の処理手順を示すフローチャートである。

【図51】

他のデバイスに関する要求の処理手順を示すフローチャートである。

【図52】

ログアウト及びヘルプに関する要求の処理手順を示すフローチャートである。

【図53】

初期メニュー画面からの処理手順を示すフローチャートである。

【図54】

公開情報の処理手順を示すフローチャートである。

【図 5 5】

ログイン処理手順を示すフローチャートである。

【図 5 6】

ジョブを選択してアクションを行なう処理手順を示すフローチャートである。

【図 5 7】

選択Jobのプレビュー処理手順を示すフローチャートである。

【図 5 8】

選択Jobの削除処理手順を示すフローチャートである。

【図 5 9】

選択Jobのスケジュール変更処理手順を示すフローチャートである。

【図 6 0】

選択Jobの印刷処理手順を示すフローチャートである。

【図 6 1】

選択Jobの印刷設定の変更処理手順を示すフローチャートである。

【図 6 2】

選択Jobの送信処理手順を示すフローチャートである。

【図 6 3】

選択Jobの送信設定の変更処理手順を示すフローチャートである。

【図 6 4】

メインメニューからの処理手順を示すフローチャートである。

【図 6 5】

検索処理手順を示すフローチャートである。

【図 6 6】

公開情報への追加処理手順を示すフローチャートである。

【図 6 7】

印刷処理手順を示すフローチャートである。

【図 6 8】

プリンタによるリモートプリント処理手順を示すフローチャートである。

【図 6 9】

送信処理手順を示すフローチャートである。

【図 7 0】

送信設定メニューの例を示す図である。

【図 7 1】

送信設定変更メニューの例を示す図である。

【図 7 2】

プリンタによるリモートセンド処理手順を示すフローチャートである。

【図 7 3】

削除したいJobの存在する装置を選択するためのメニューの例を示す図である

【図 7 4】

削除Jobのリストの例を示す図である。

【図 7 5】

削除処理手順を示すフローチャートである。

【図 7 6】

プリンタによるリモートデリート処理手順を示すフローチャートである。

【図 7 7】

命令の設定処理手順を示すフローチャートである。

【図 7 8】

スケジュール変更処理手順を示すフローチャートである。

【図 7 9】

リモートリストケジュール処理手順を示すフローチャートである。

【図 8 0】

他のデバイスへのアクセス処理手順を示すフローチャートである。

【図 8 1】

他のデバイスにアクセスするための設定画面を示す図である。

【図 8 2】

デスクトップの処理手順を示すフローチャートである。

【図 8 3】

未読メール処理手順を示すフローチャートである。

【図 8 4】

既読メール処理手順を示すフローチャートである。

【図 8 5】

未読メールのリストの例を示す図である。

【図 8 6】

メールの内容の表示例を示す図である。

【図 8 7】

既読メールのリストの例を示す図である。

【図 8 8】

ファイル処理の処理手順を示すフローチャートである。

【図 8 9】

ResetInstructionsの処理手順を示すフローチャートである。

【図 9 0】

DeleteAllInstructionsの処理手順を示すフローチャートである。

【図 9 1】

DeleteInstructionsの処理手順を示すフローチャートである。

【図 9 2】

GetInstructionDetailsの処理手順を示すフローチャートである。

【図 9 3】

Checkの処理手順を示すフローチャートである。

【図 9 4】

GetJobDataの処理手順を示すフローチャートである。

【図 9 5】

GetJobAttributesの処理手順を示すフローチャートである。

【図 9 6】

GetActionAttributesの処理手順を示すフローチャートである。

【図 9 7】

DeleteJobの処理手順を示すフローチャートである。

【図 9 8】

GetNotificationListの処理手順を示すフローチャートである。

【図 9 9】

SetNotificationConditionの処理手順を示すフローチャートである。

【図 1 0 0】

SetNotificationConditionの処理手順を示すフローチャートである。

【図 1 0 1】

SetNotificationStatusの処理手順を示すフローチャートである。

【図 1 0 2】

ClearHistoryの処理手順を示すフローチャートである。

【図 1 0 3】

DeleteAllPublicInfoJobsの処理手順を示すフローチャートである。

【図 1 0 4】

PrintPublicInfoの処理手順を示すフローチャートである。

【図 1 0 5】

SearchDataの処理手順を示すフローチャートである。

【図 1 0 6】

初期メニューの画面例を示す図である。

【図 1 0 7】

URLの印刷設定画面を示す図である。

【図 1 0 8】

パスワード入力画面を示す図である。

【図 1 0 9】

プリントキュー内で選択されたジョブに対する処理の設定画面を示す図である

【図 1 1 0】

公開情報の処理設定画面を示す図である。

【図111】

情報を他のデバイス上で公開するための設定画面の例を示す図である。

【図112】

ログイン画面を示す図である。

【図113】

保留されたジョブのリストの例を示す図である。

【図114】

ジョブのプレビューの画面表示例を示す図である。

【図115】

削除の確認画面を示す図である。

【図116】

スケジュールの変更のための設定画面を示す図である。

【図117】

印刷設定メニューの例を示す図である。

【図118】

通知設定画面の例を示す図である。

【図119】

印刷設定変更メニューの例を示す図である。

【図120】

修正メニューの例を示す図である。

【図121】

メインメニューの例を示す図である。

【図122】

印刷対象のジョブのある場所を選択するためのメニューの例を示す図である。

【図123】

印刷対象のジョブを選択するためのメニューの例を示す図である。

【図124】

送信対象のジョブのある場所を選択するためのメニューの例を示す図である。

【図125】

送信対象のジョブを選択するためのメニューの例を示す図である。

【図126】

命令設定画面の例を示す図である。

【図127】

変更したいJobの存在する装置を選択するためのメニューの例を示す図である

【図128】

スケジュールの変更対象Jobの選択画面の例を示す図である。

【図129】

デスクトップメインメニューの例を示す図である。

【図130】

ファイル取得のための設定画面の例を示す図である。

【図131】

デスクトップのファイルの一覧表示の例を示す図である。

【図132】

取得ファイルに対する処理の選択画面の例を示す図である。

【図133】

ジョブに対する制限の設定のための設定画面を示す図である。

【図134】

ジョブの拒絶の設定のための設定画面を示す図である。

【符号の説明】

101 ユーザ

102 クライアントコンポーネント

103 サーバコンポーネント

104 データベース

105 デーモン

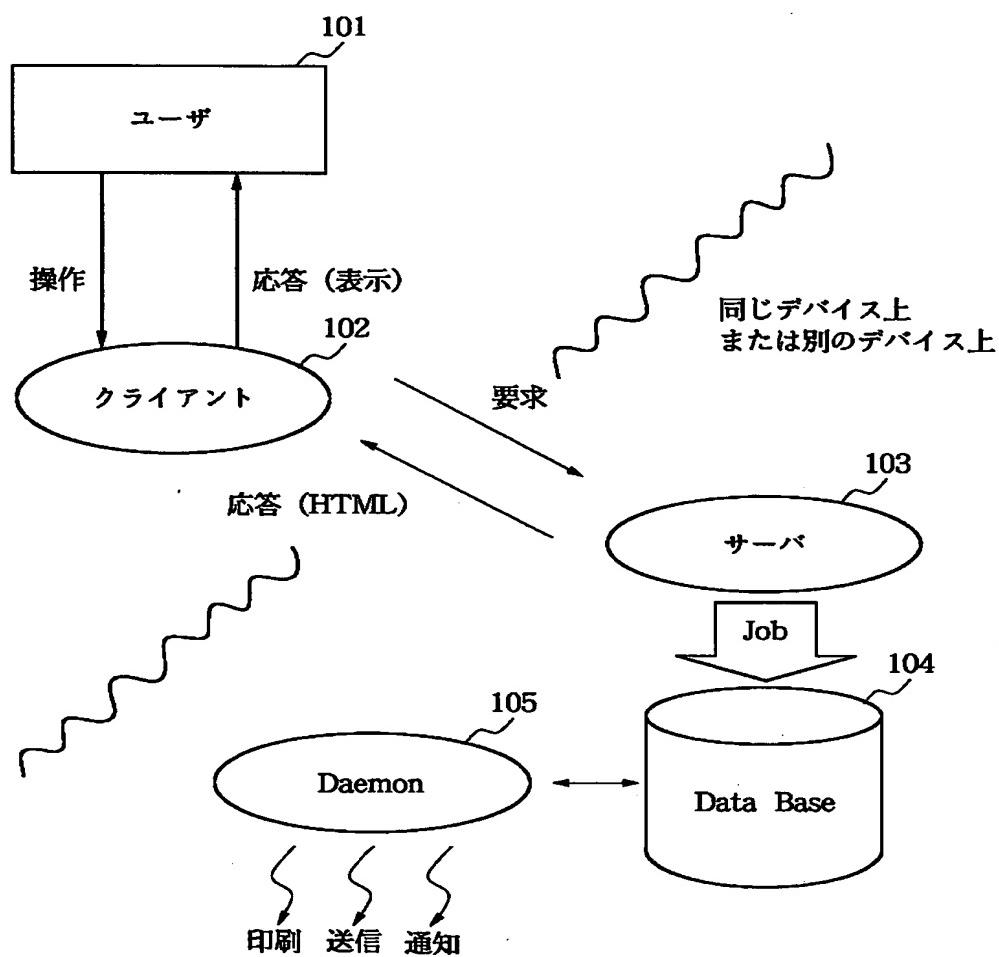
201 操作部

202、203 ウェブブラウザ

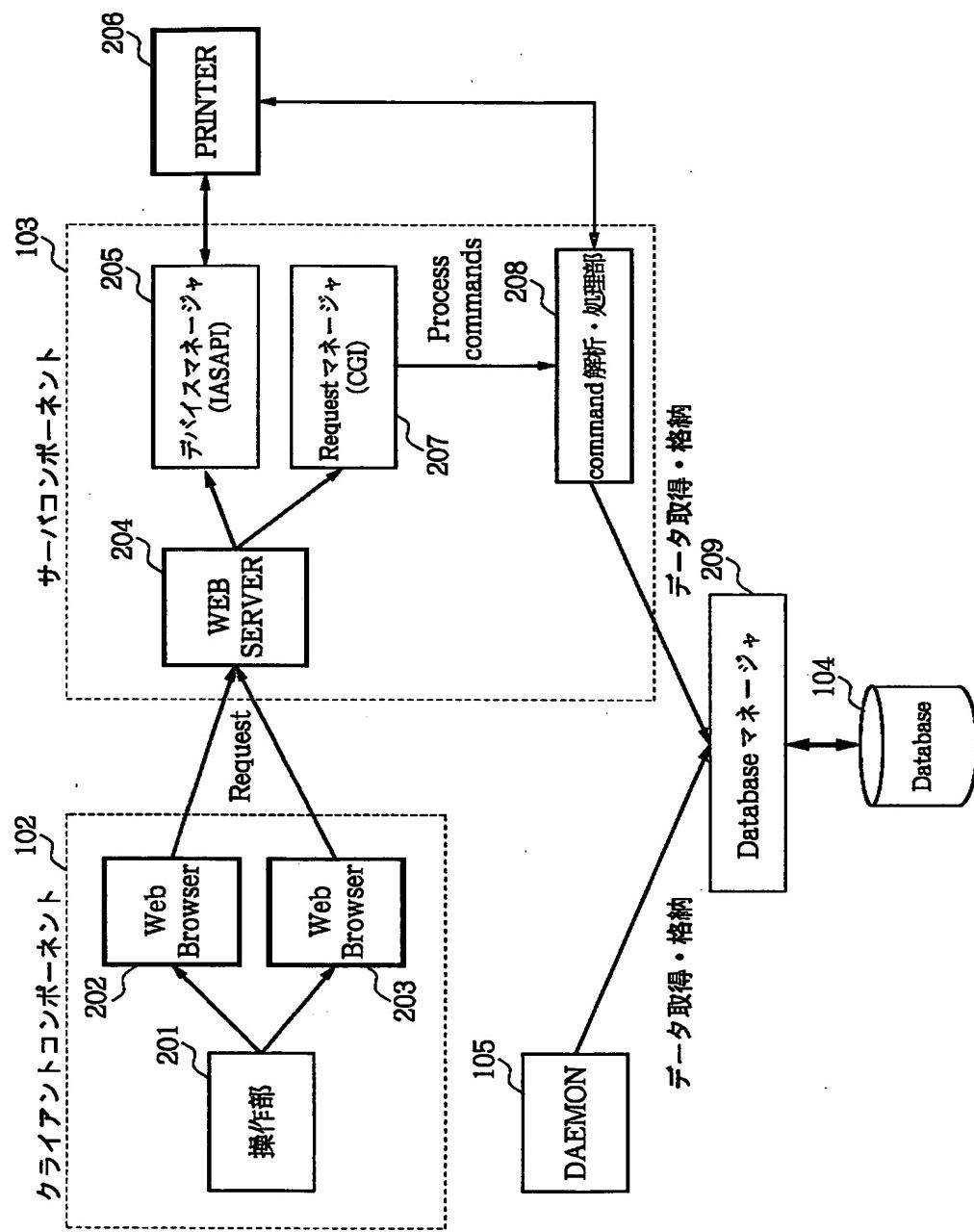
- 204 ウェブサーバ
- 205 デバイスマネージャ
- 206 プリンタ
- 207 リクエストマネージャ
- 208 コマンド解析部
- 209 データベースマネージャ
- 601 印刷制御部
- 602 制御部
- 603 変換制御部
- 604 表示制御部
- 605 通知制御部
- 606 外部アプリ制御部
- 607 印刷部
- 608 メール制御部
- 701 ネットワーク
- 702 プリンタ
- 703 入出力操作部
- 704 スキャナ
- 705 マルチファンクション機器
- 706 パーソナルコンピュータ
- 801 タッチパネル
- 802 C P U
- 803 印刷部
- 804 通信部
- 805 R O M
- 806 R A M
- 807 H D D
- 808 システムバス

【書類名】 図面

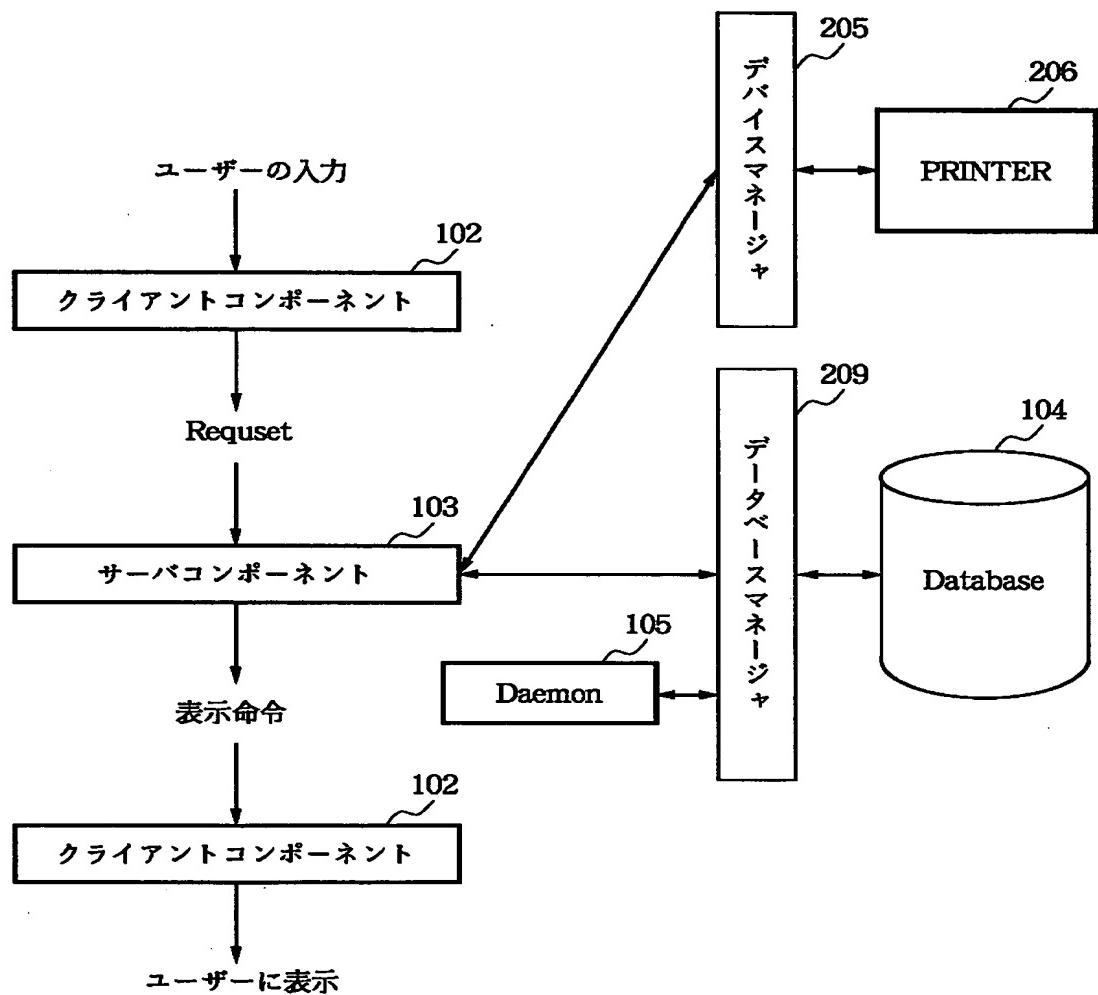
【図1】



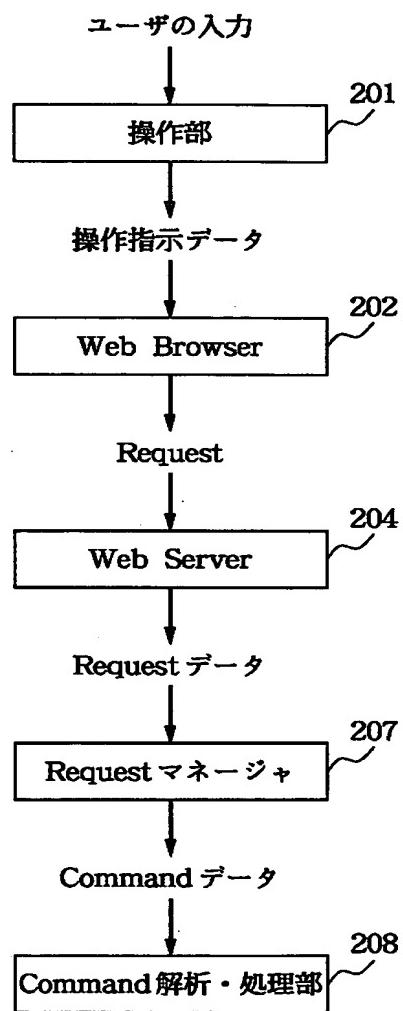
【図2】



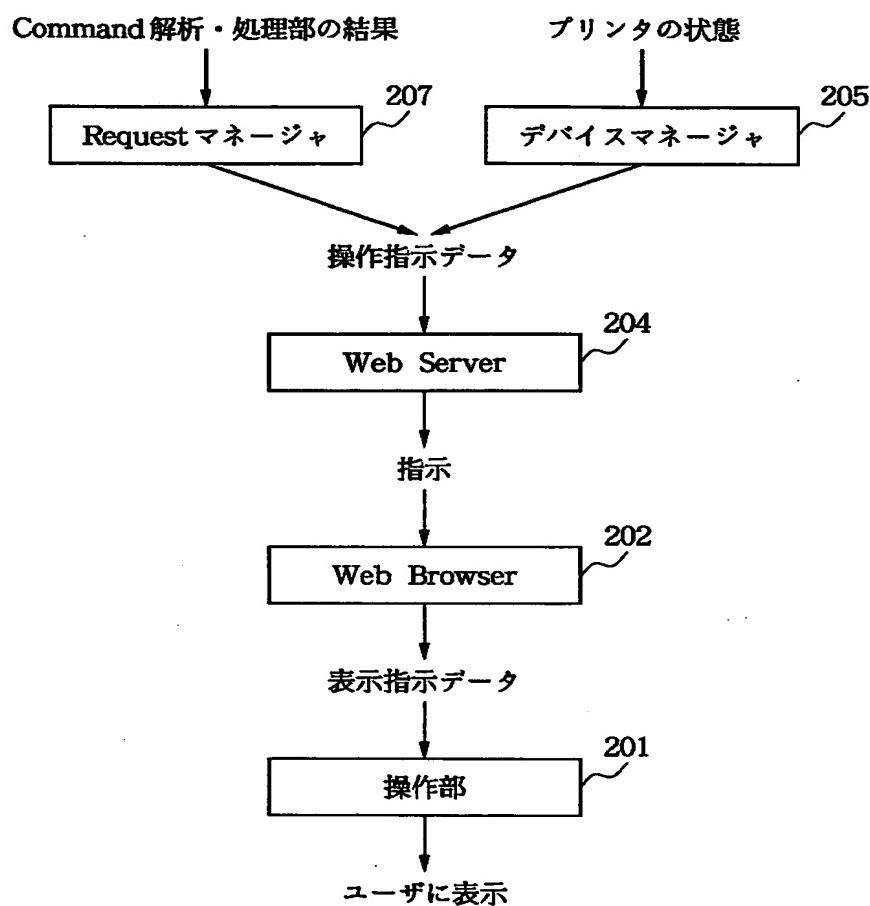
【図3】



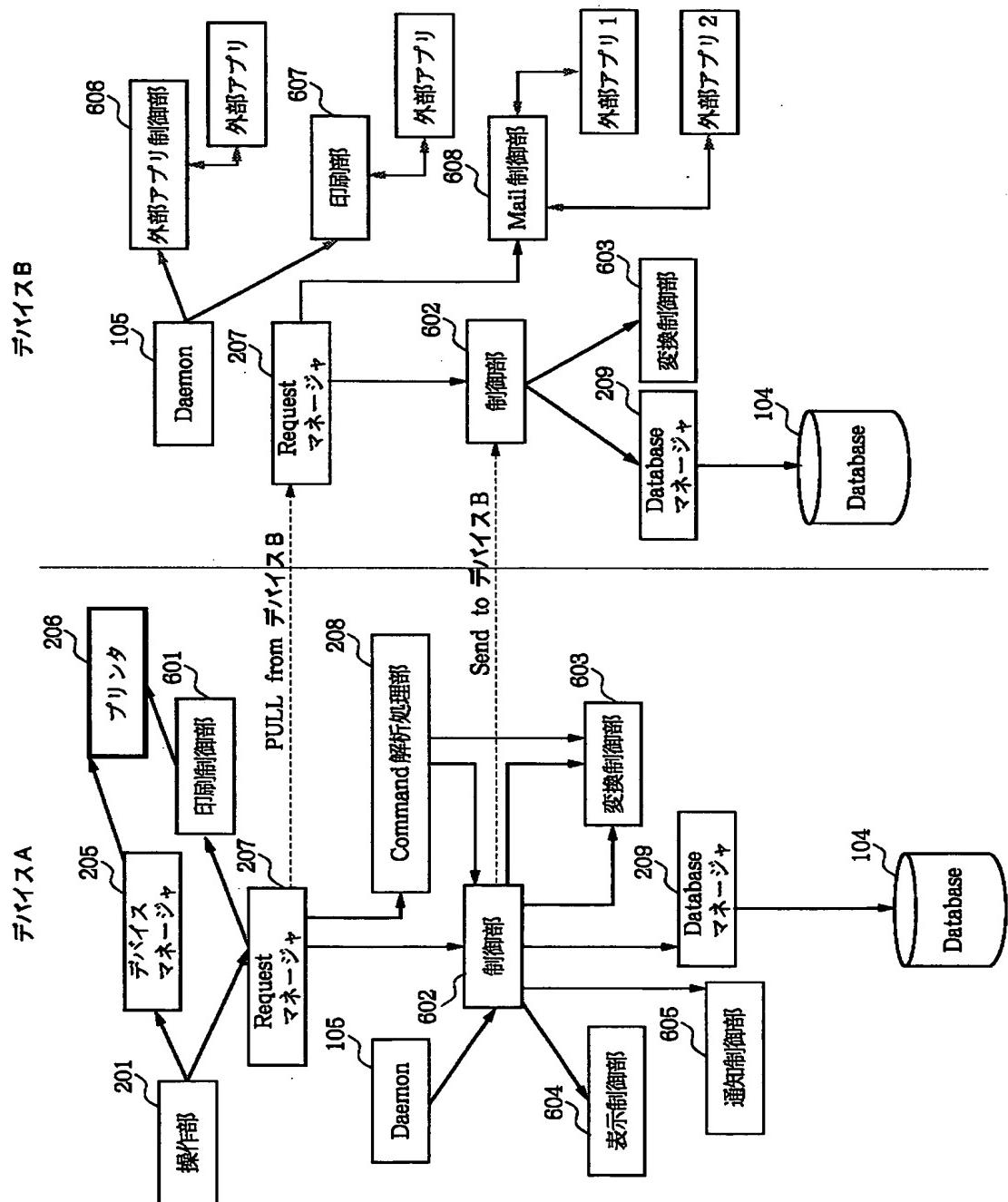
【図4】



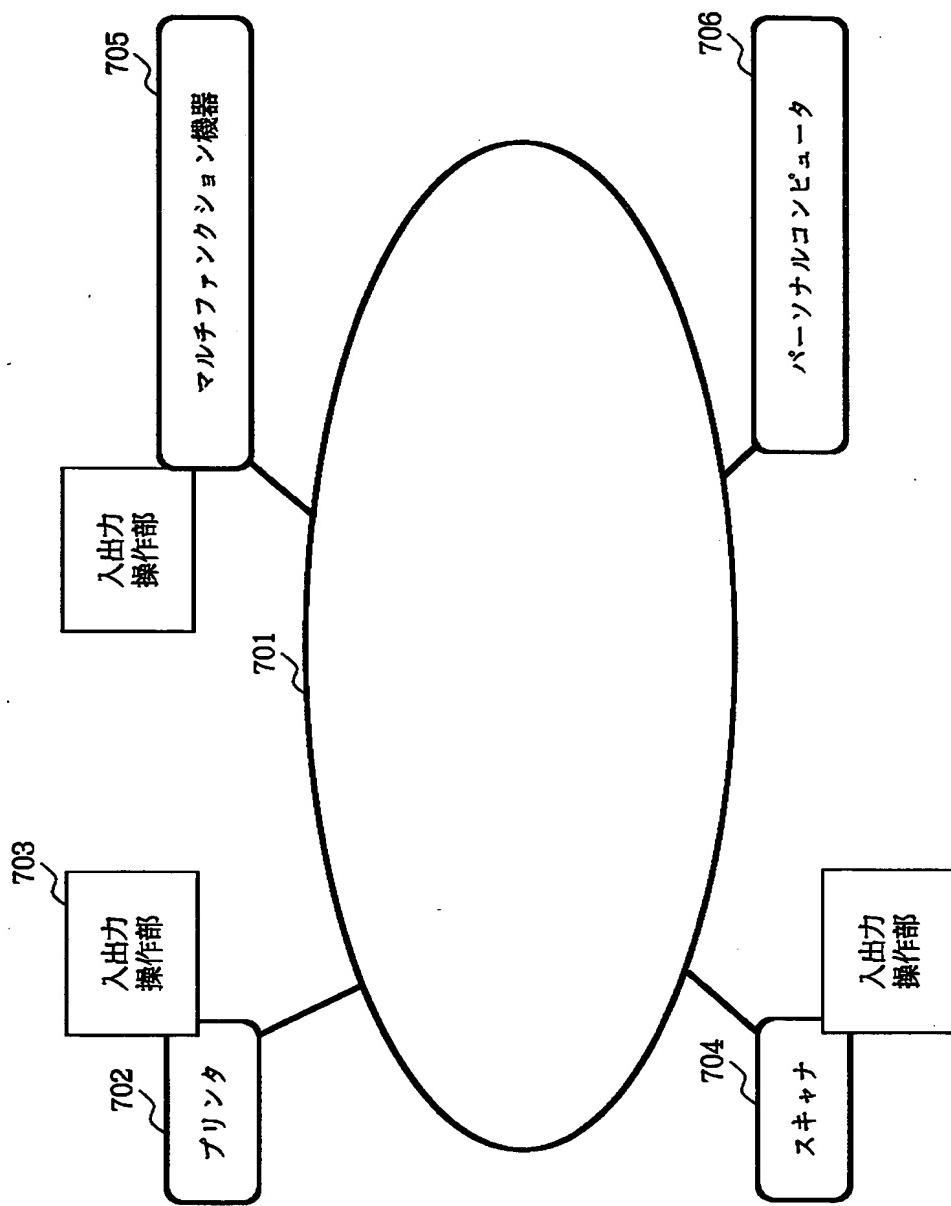
【図5】



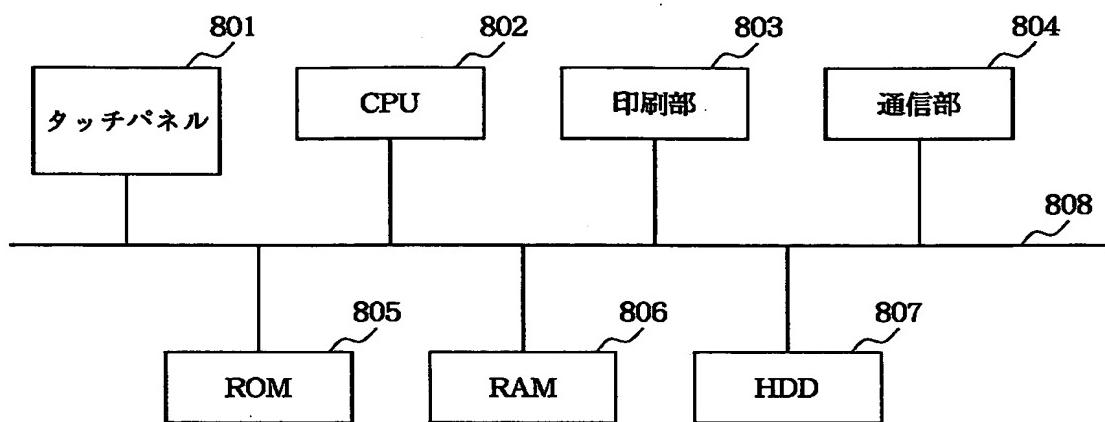
【図6】



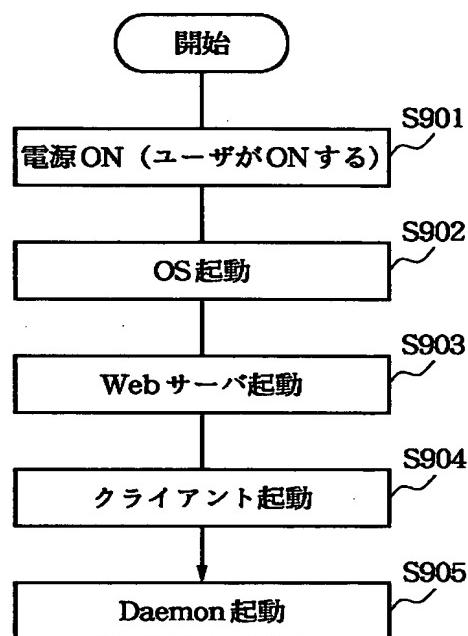
【図7】



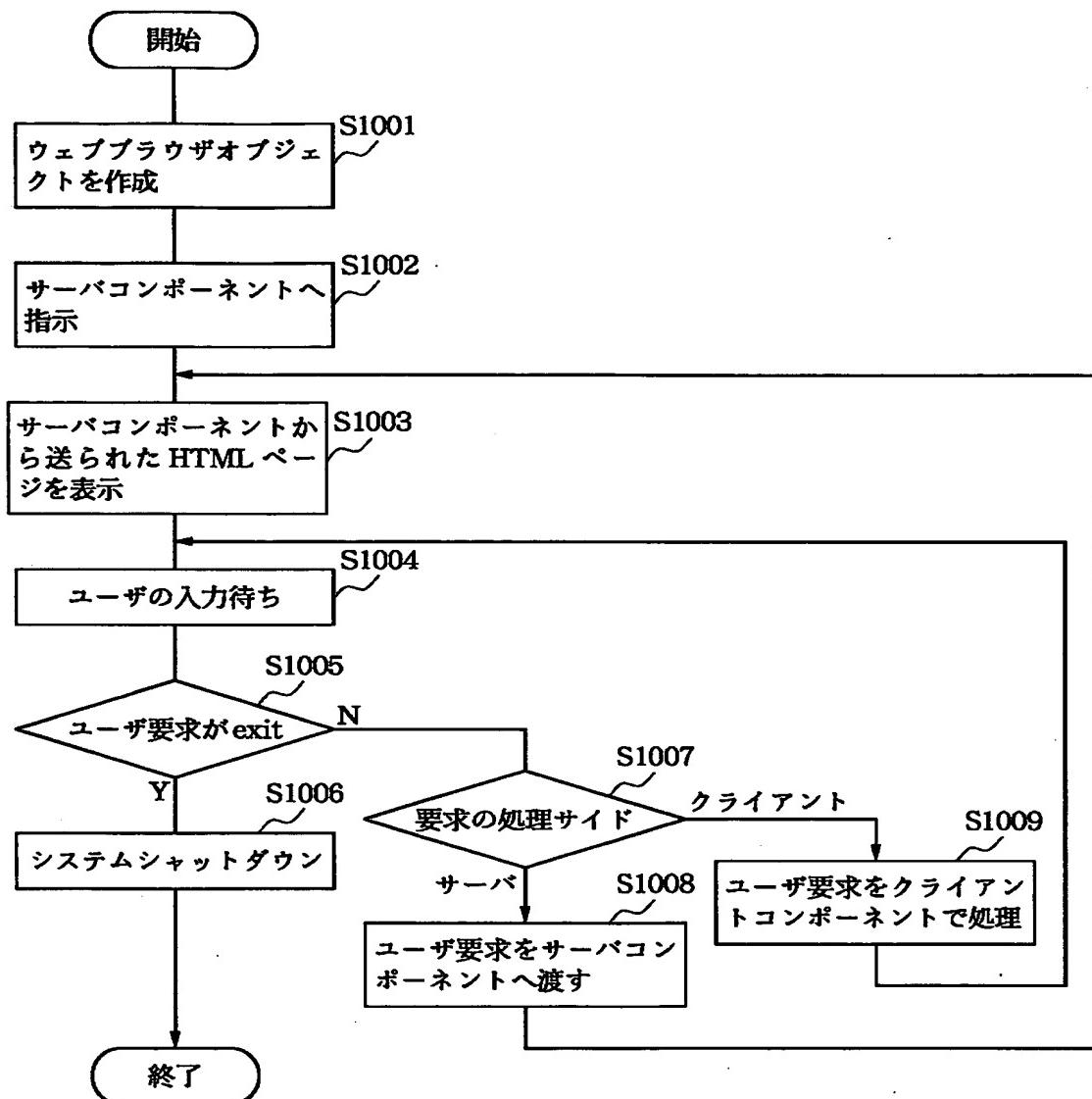
【図8】



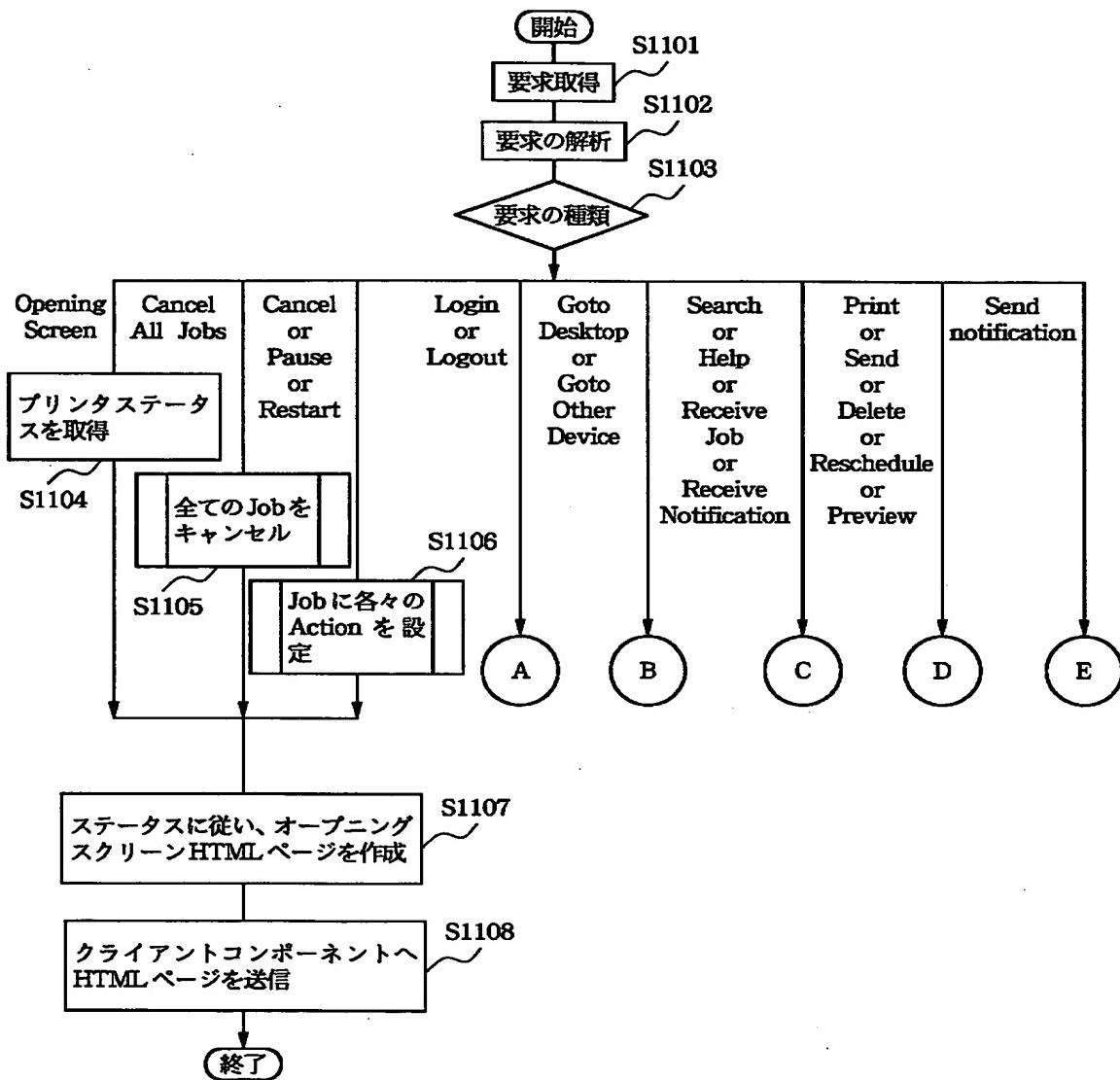
【図9】



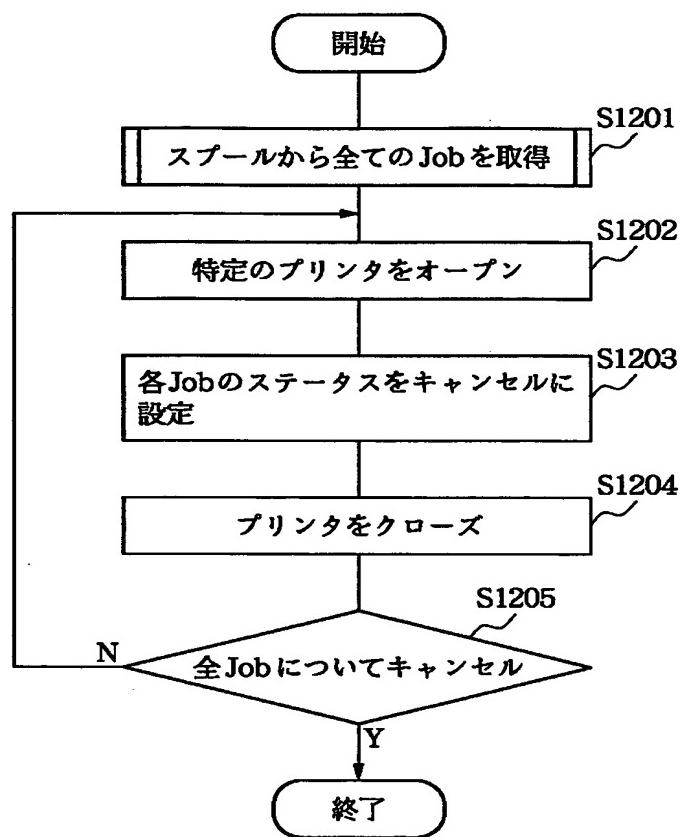
【図10】



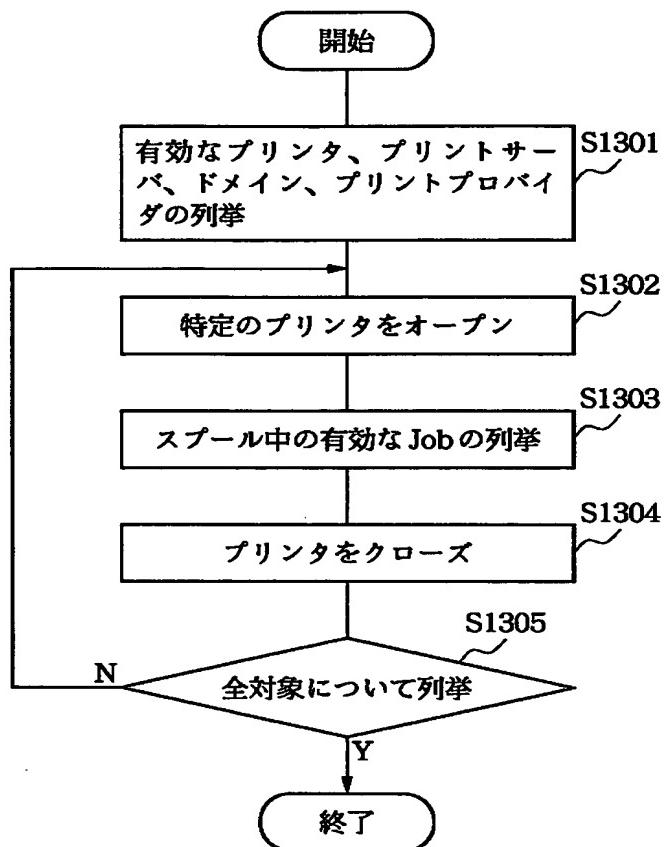
【図11】



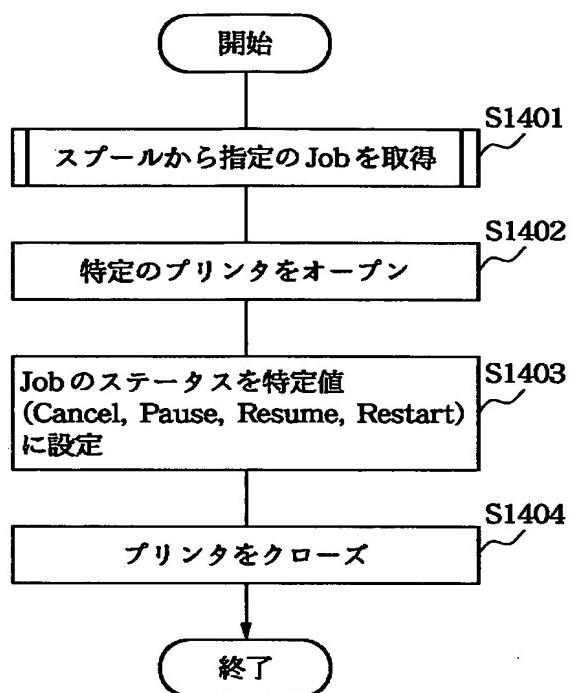
【図12】



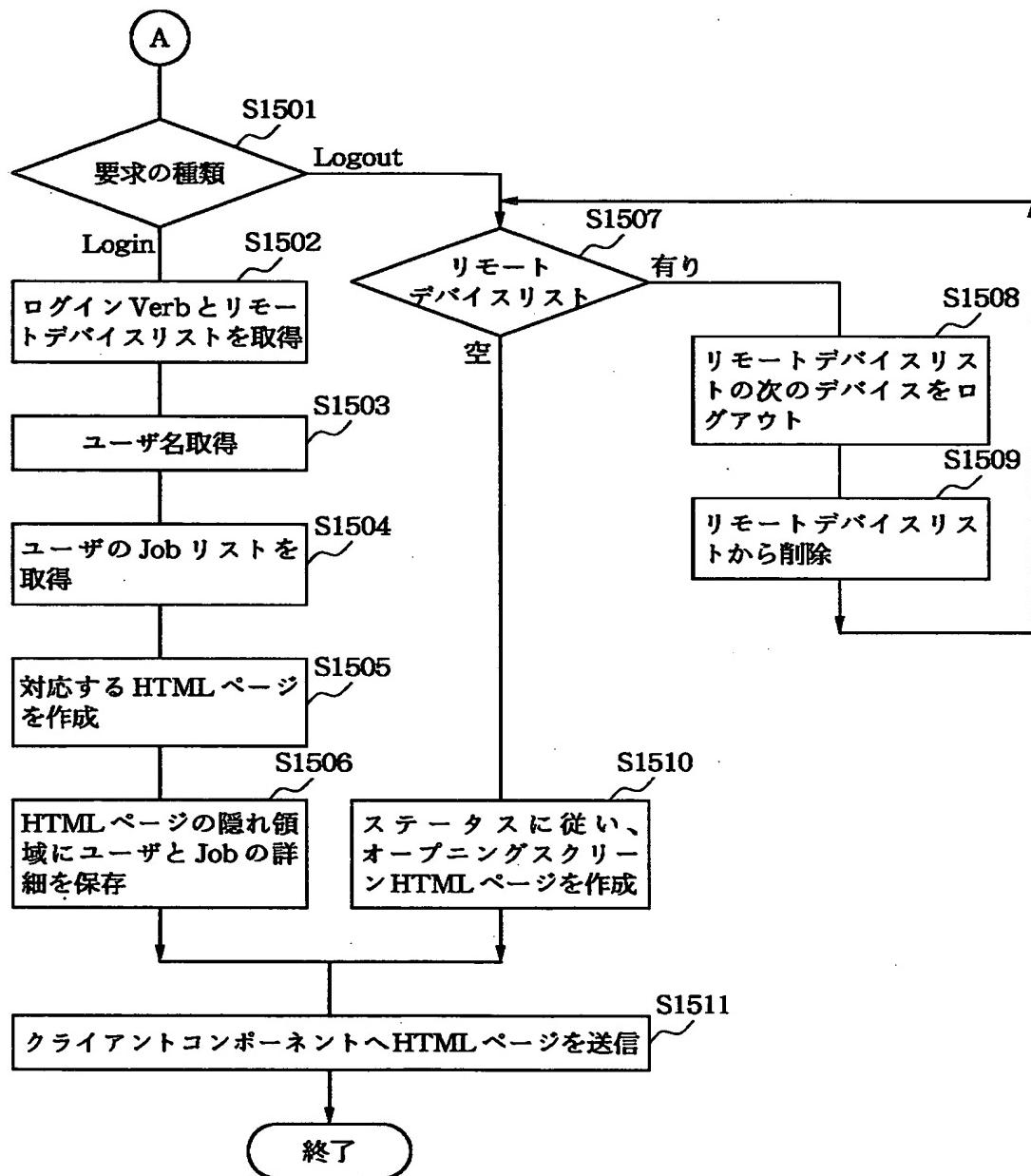
【図13】



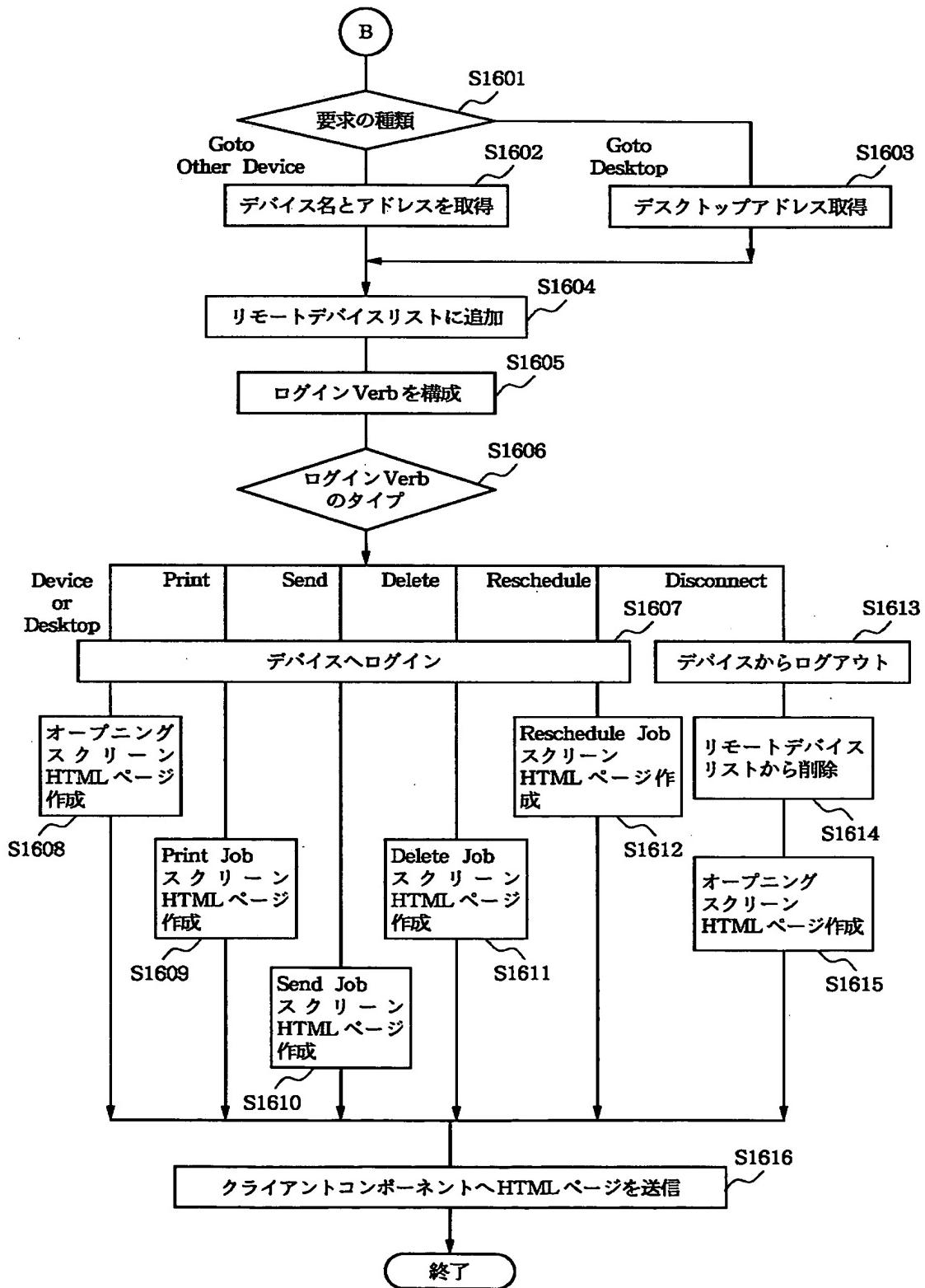
【図14】



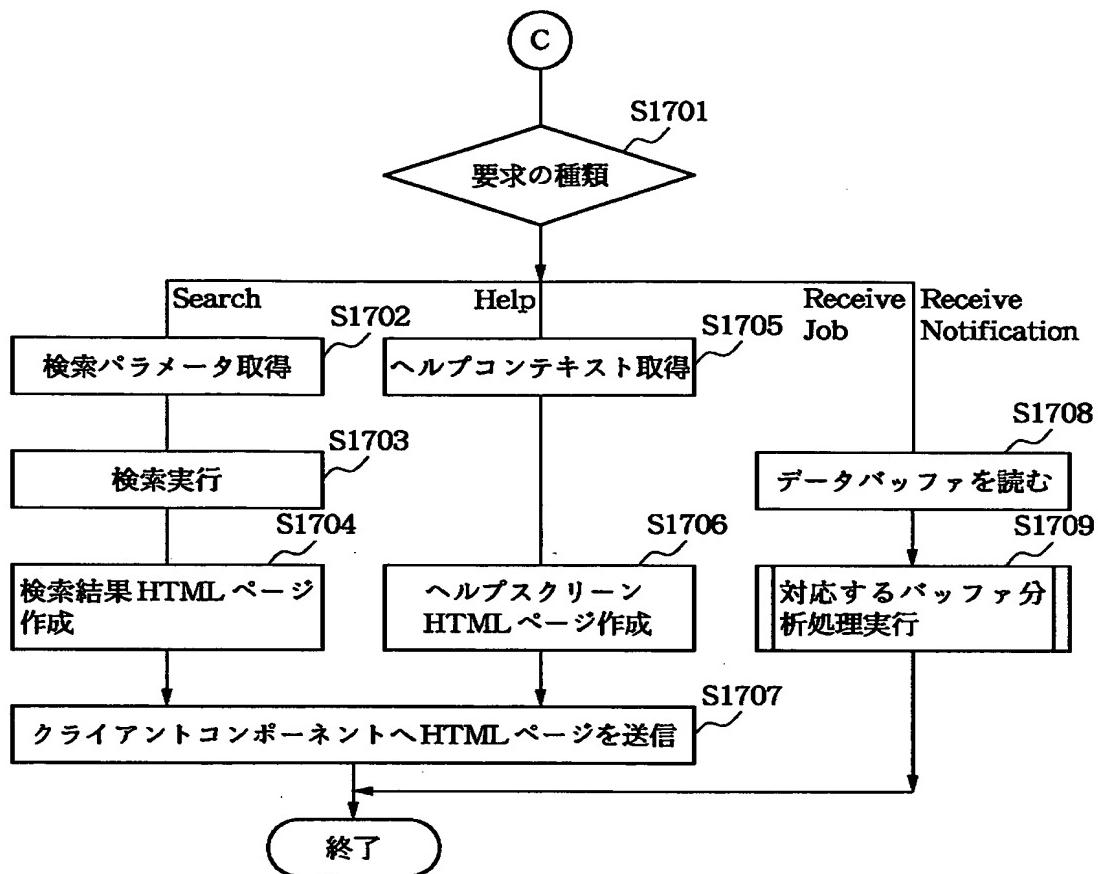
【図15】



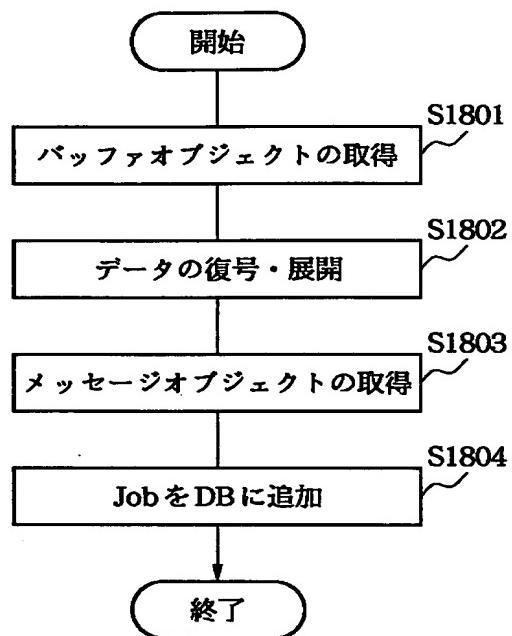
【図16】



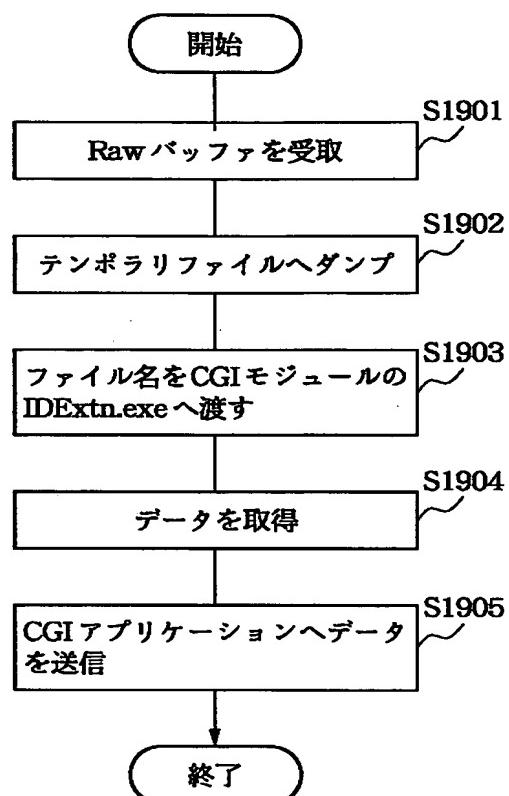
【図17】



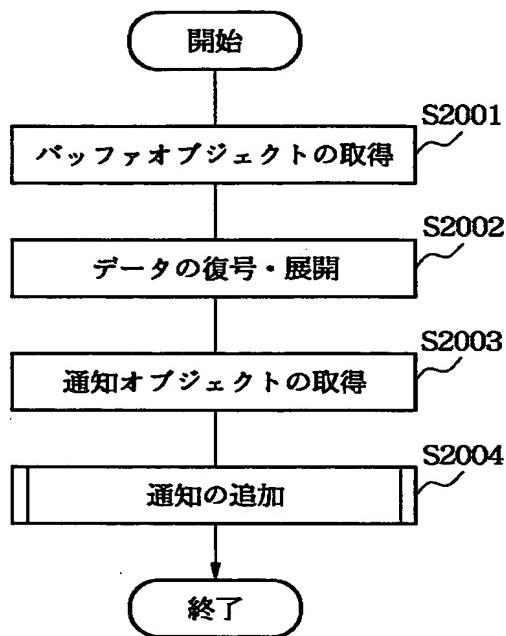
【図18】



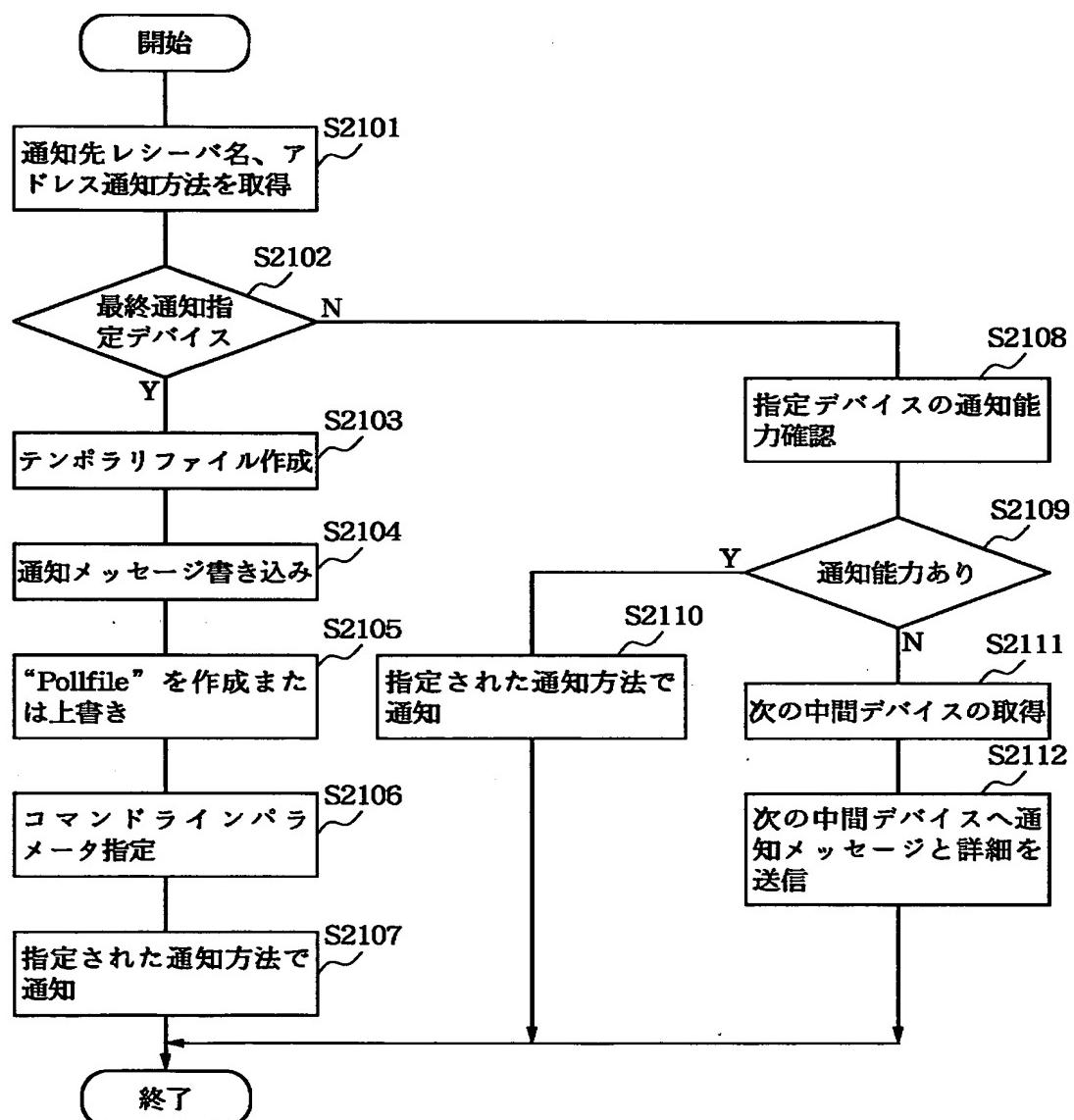
【図19】



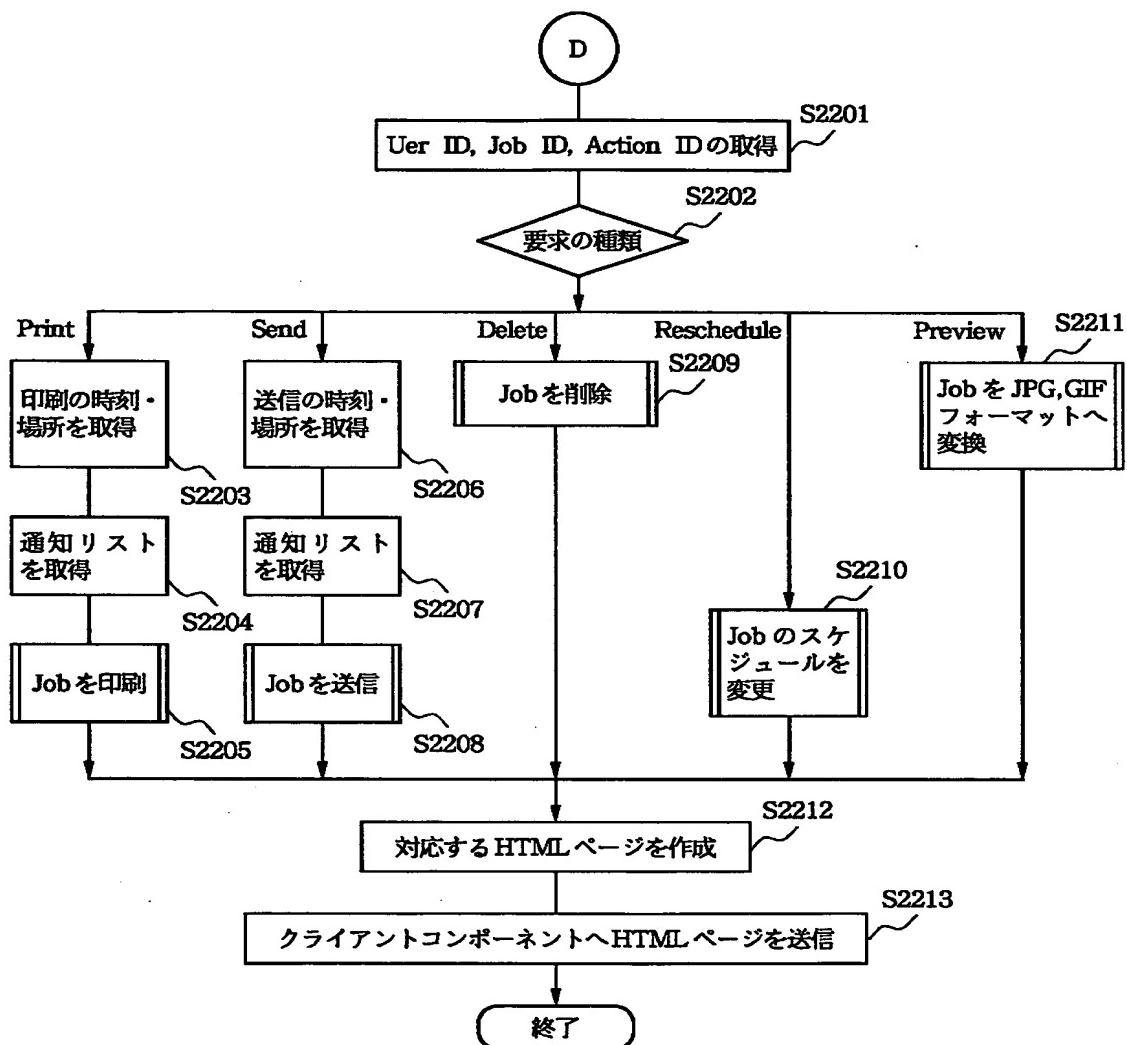
【図20】



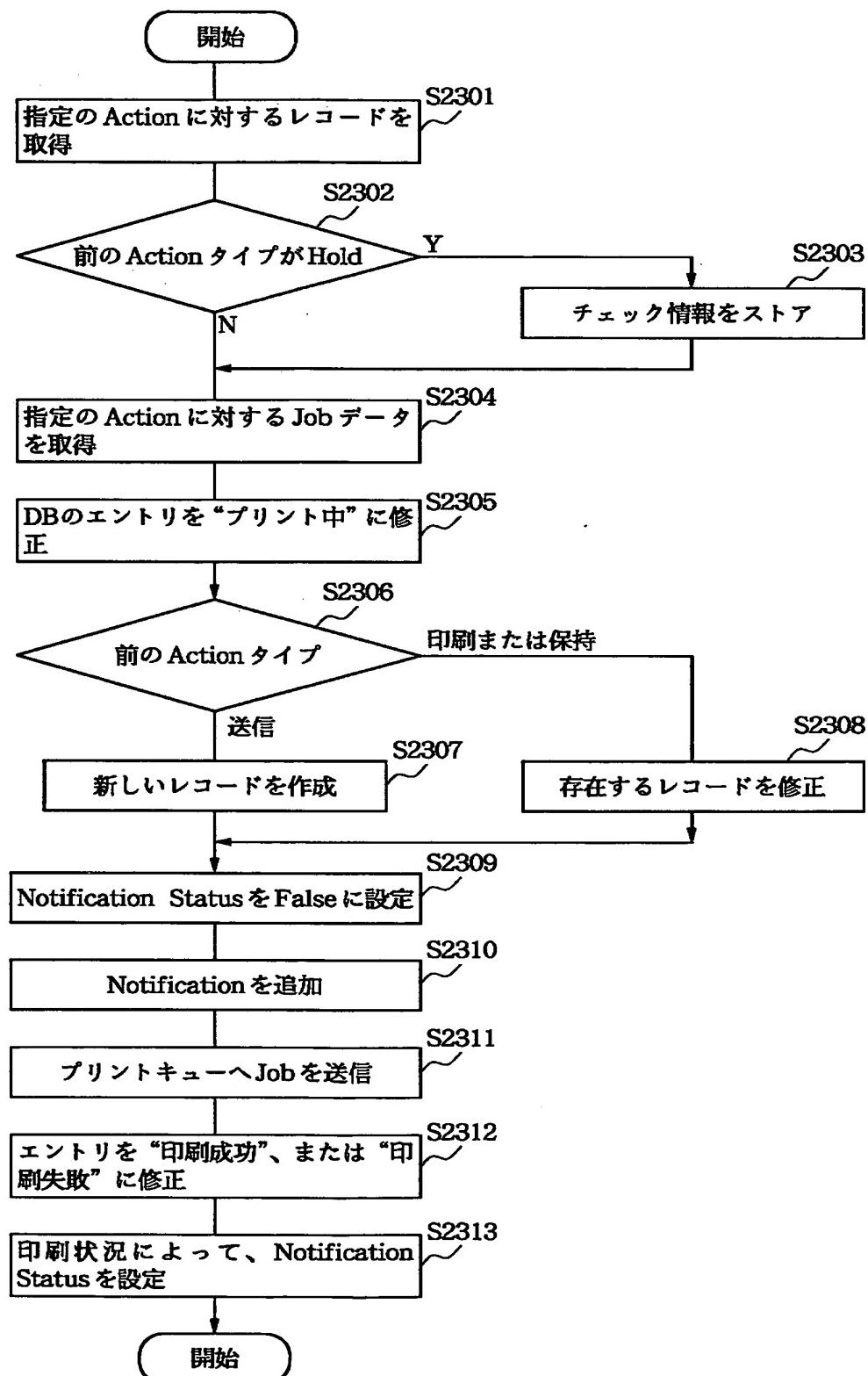
【図21】



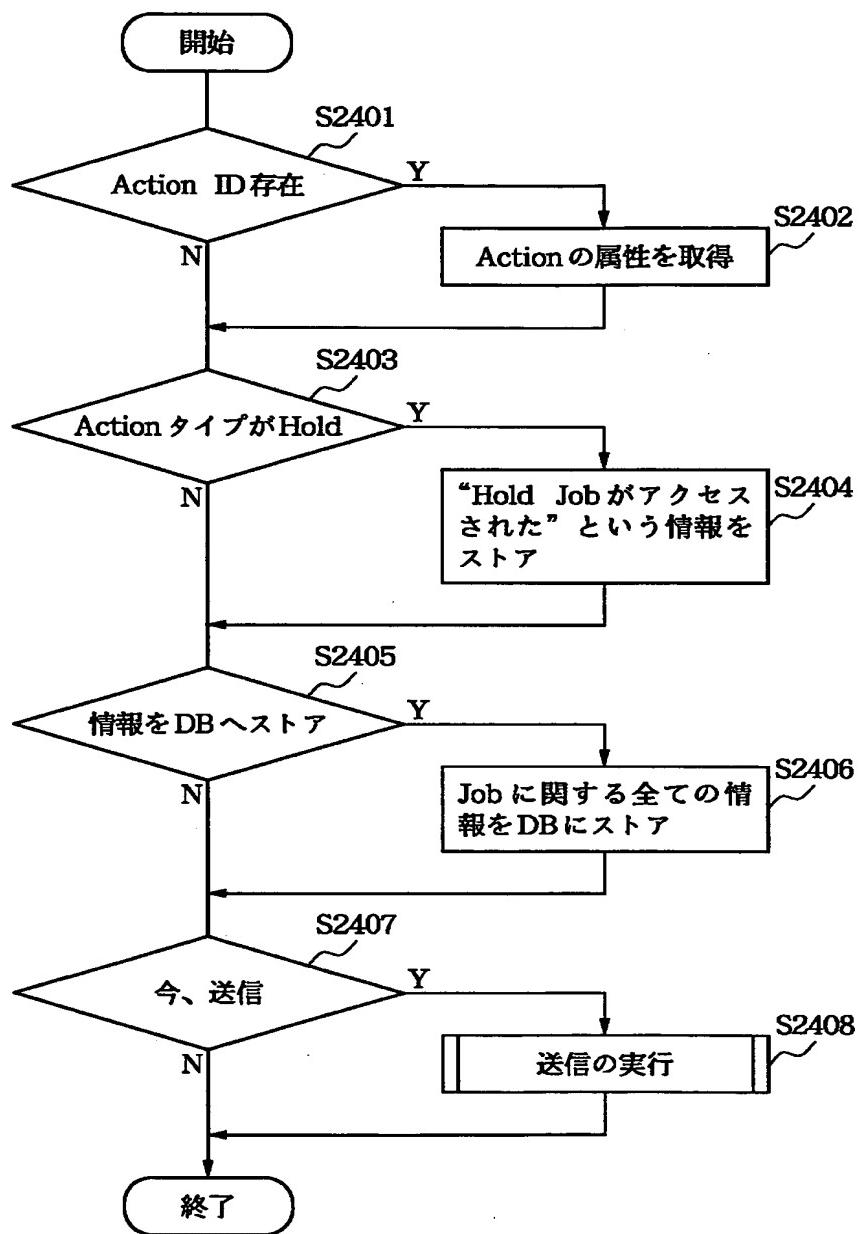
【図22】



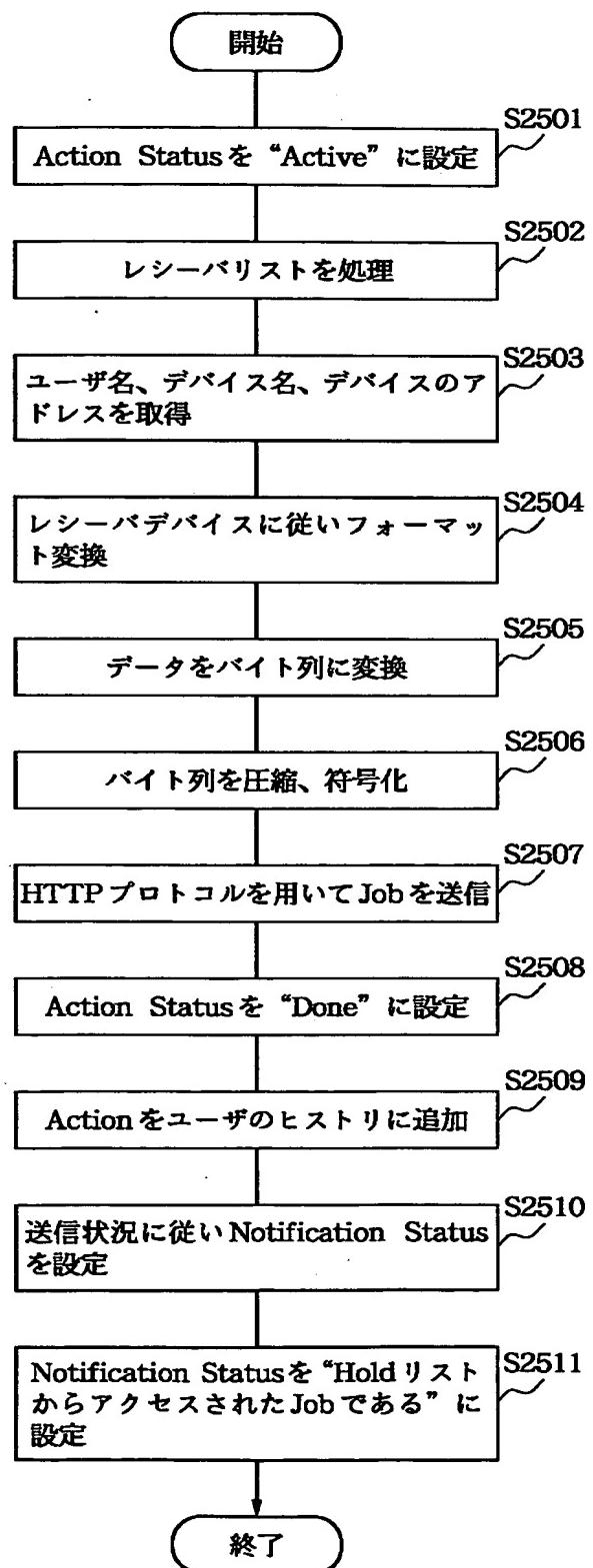
【図23】



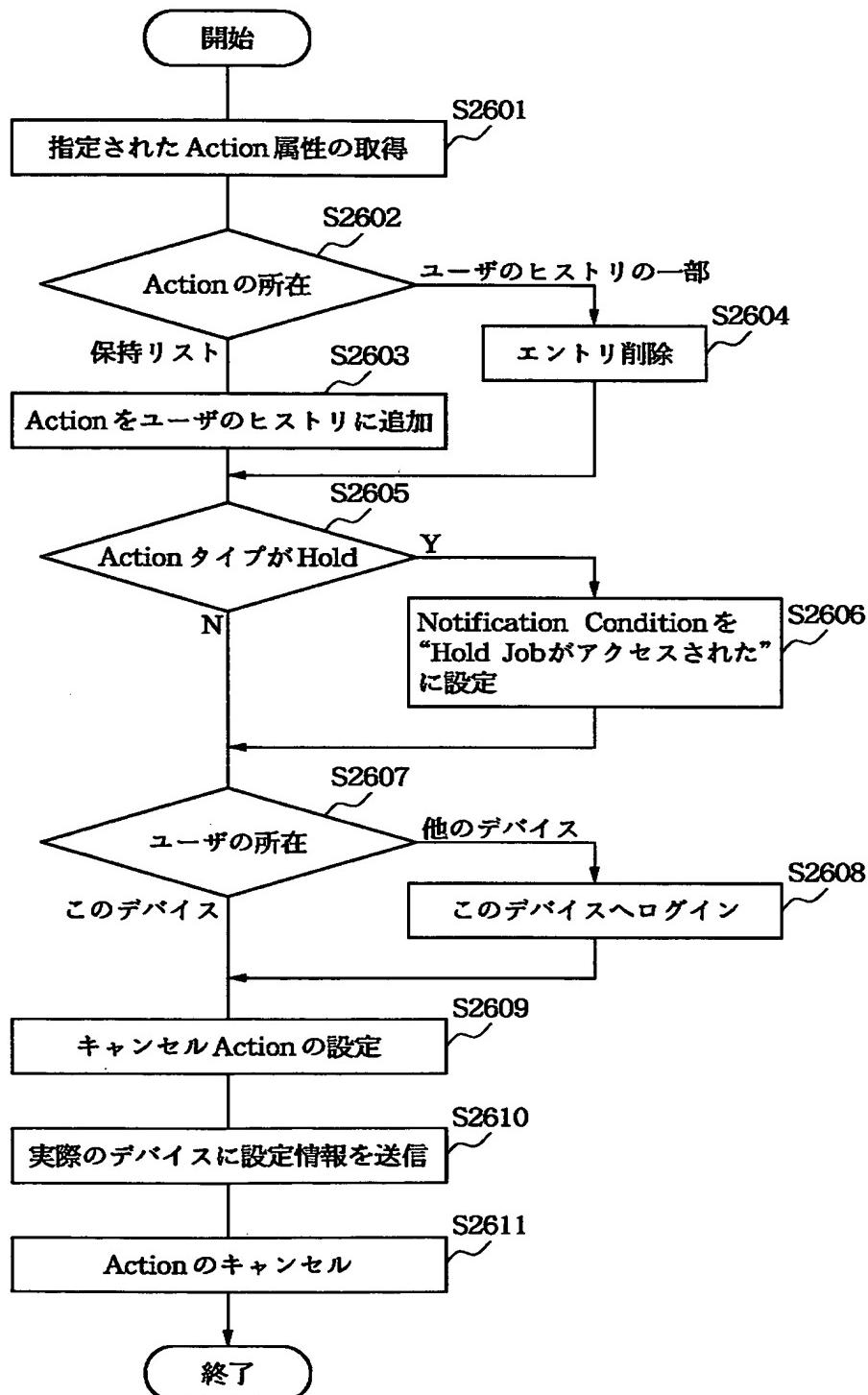
【図24】



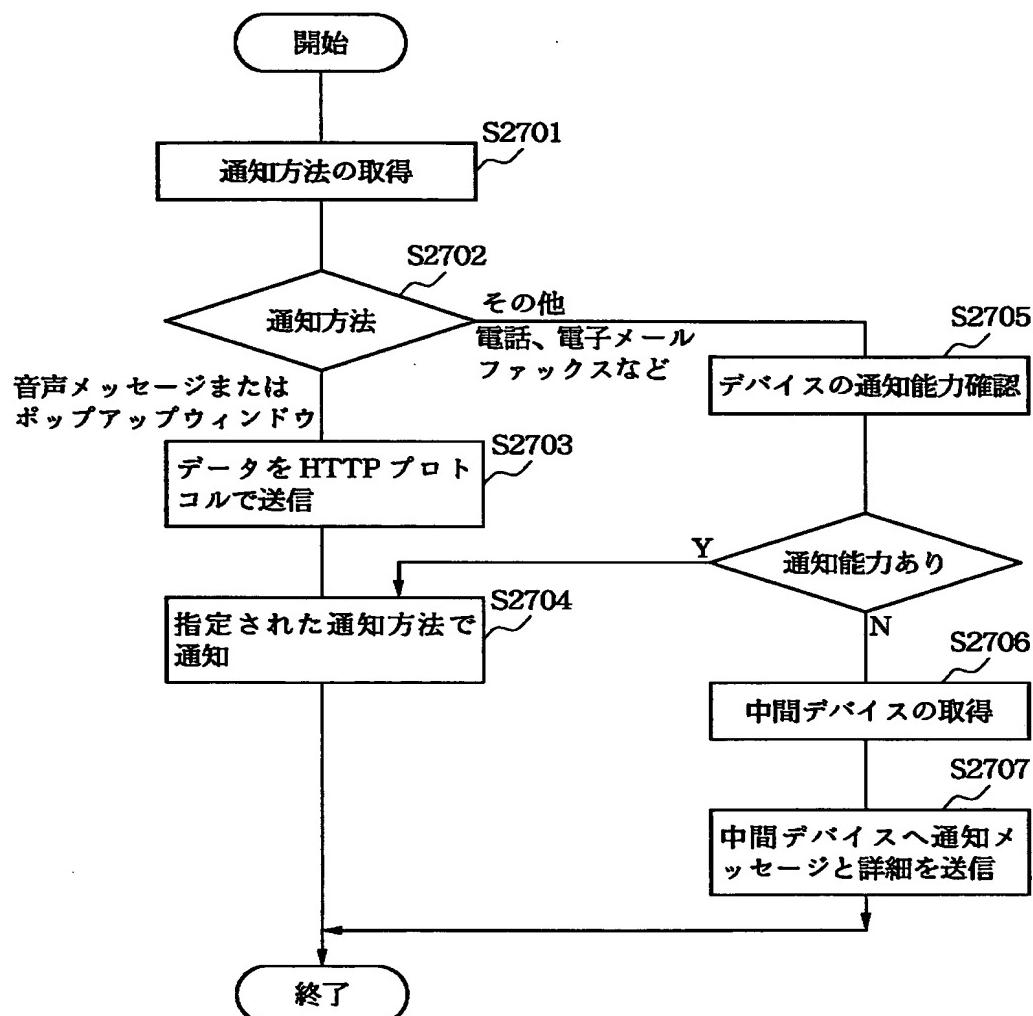
【図25】



【図26】



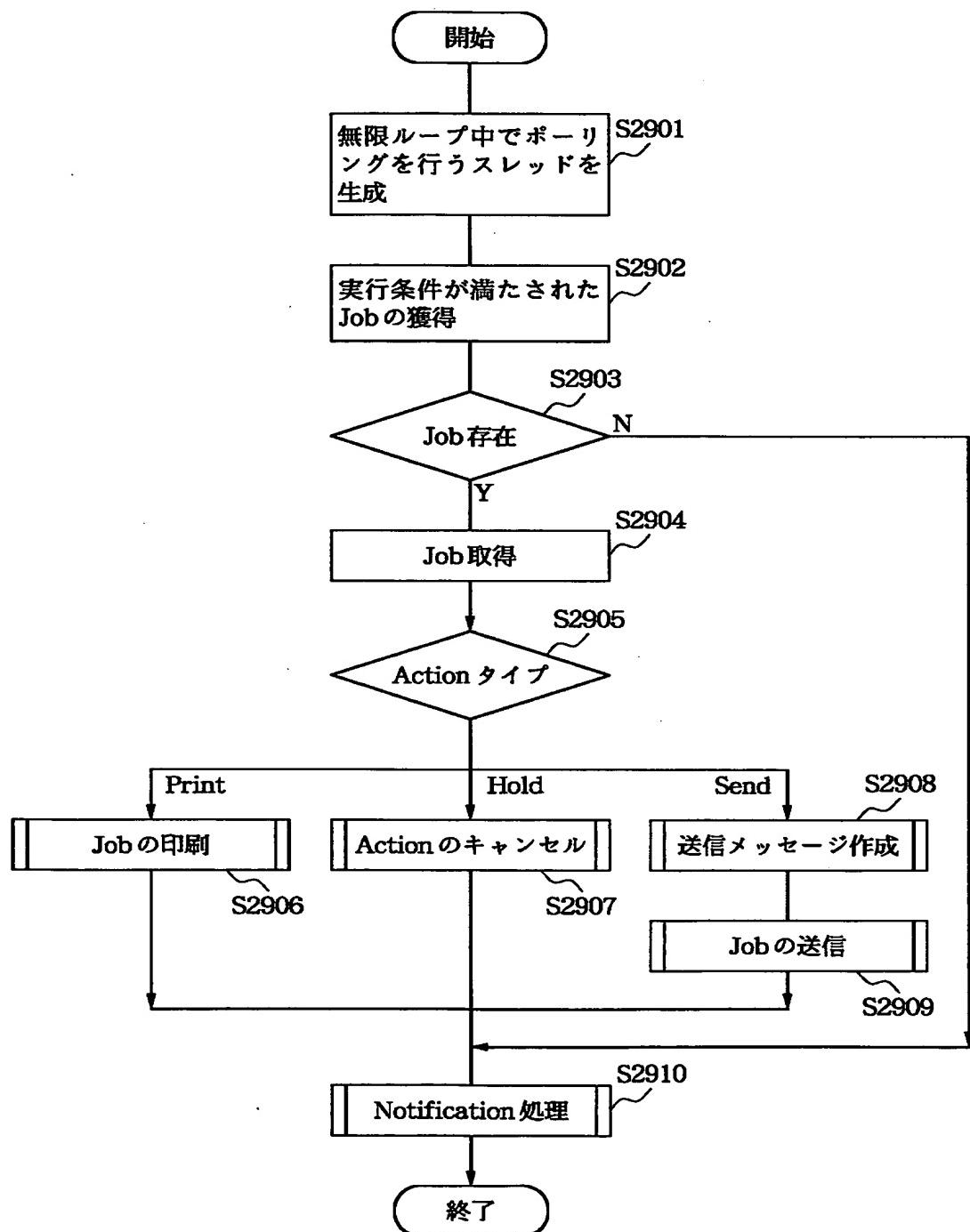
【図27】



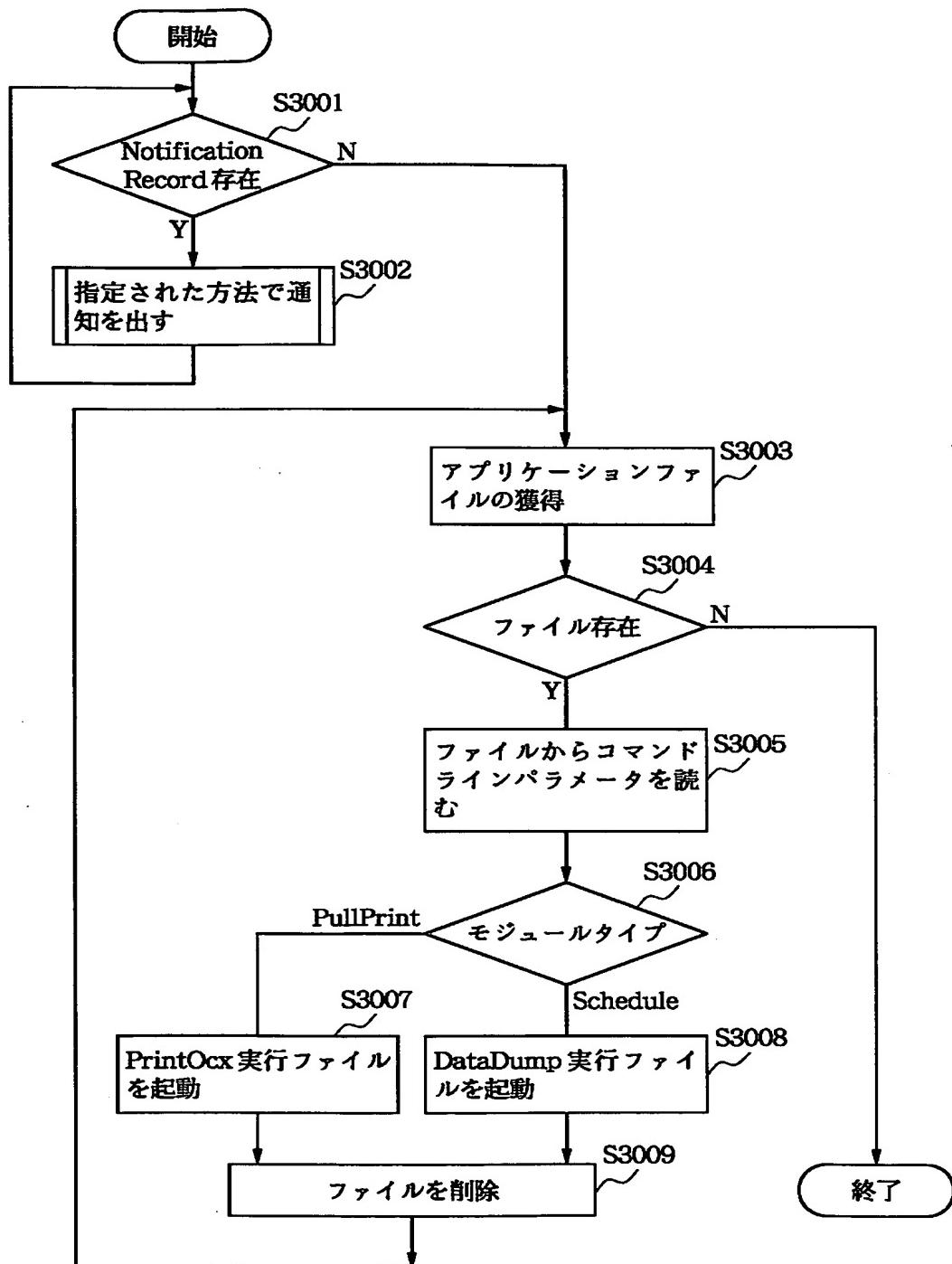
【図28】

[サーバの入力情報]	
入力の種類	処理
(要求) Print Job Send Job Receive Job Cancel All Jobs Send Notification Receive Notification Cancel/pause/Restart (Receive)	Job を印刷 Job を送信 Job をDBへ追加 全てのJobをキャンセル Notificationを出す Notificationを受取 Jobを各々のActionに設定
Preview Search Delete Reschedule Help Opening Screen Main Menu Login Logout Goto Other Device Goto Desktop	対応する新しいHTMLページ を表示

【図29】



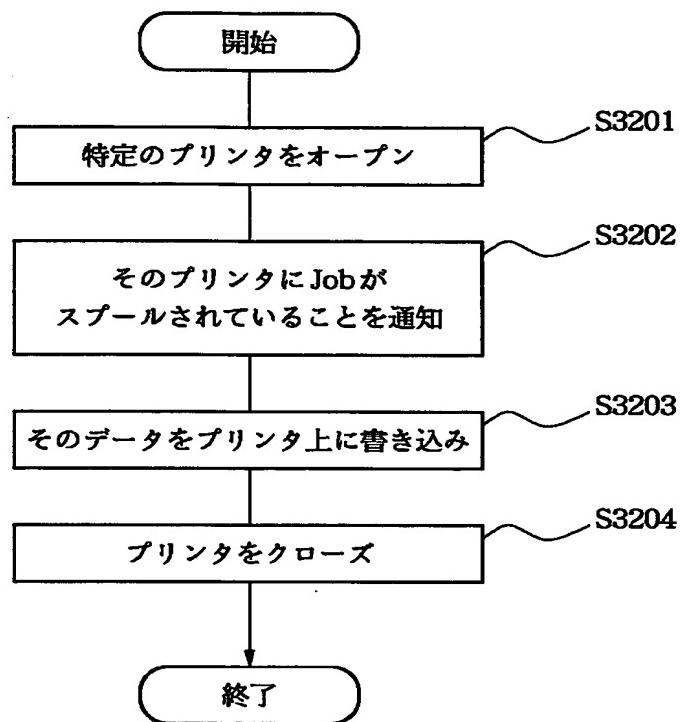
【図30】



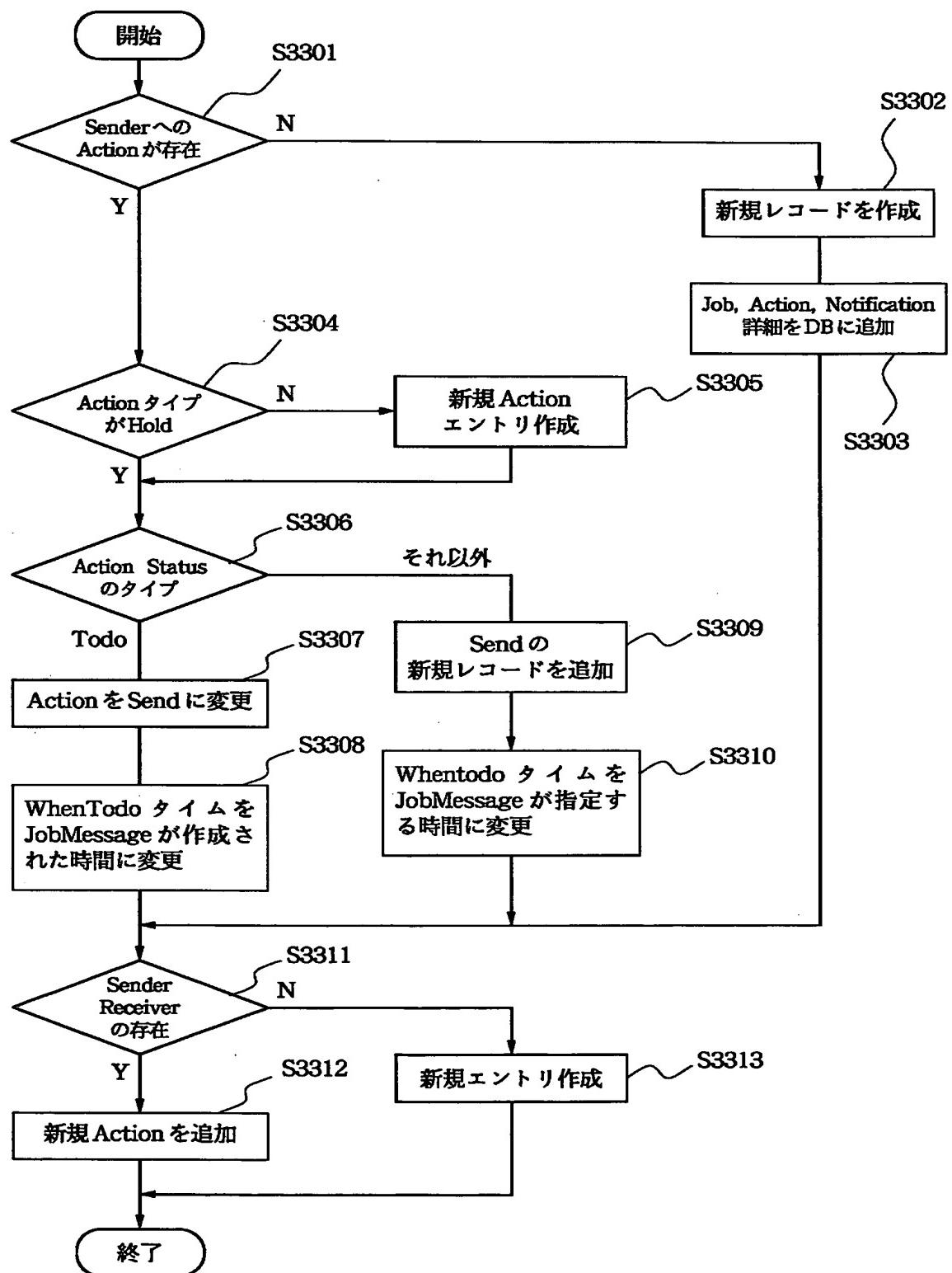
【図31】

[Job情報]	
Action Type	
Print	対象ファイルを印刷する
Send	対象ファイルを送信する
Hold	対象Jobを取り消す
Job対象ファイル	
Print File	
Send File	
Hold File	
Notification	
有り／無し	
時刻	
起動時刻	実行／記憶期限
有効期限	

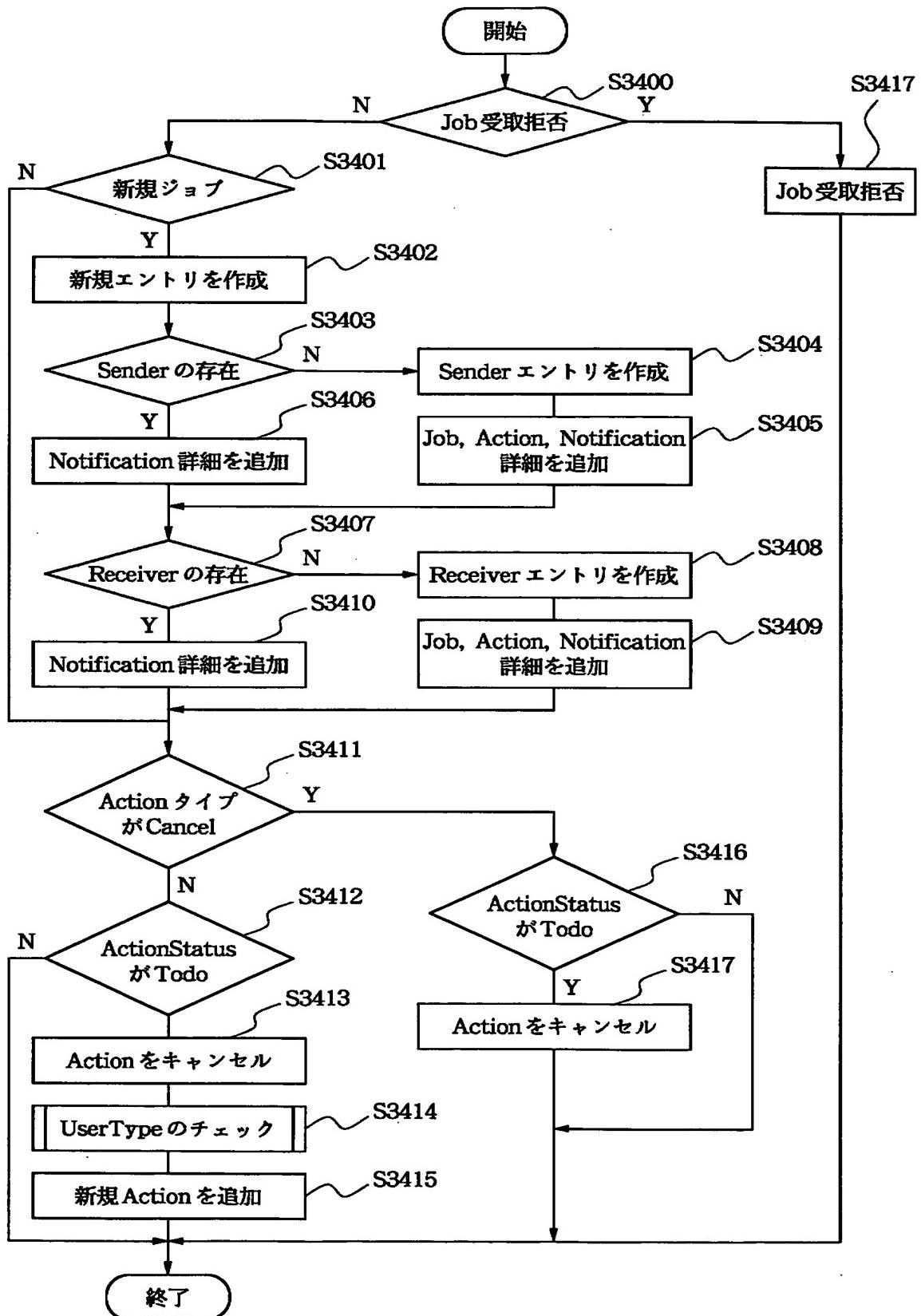
【図32】



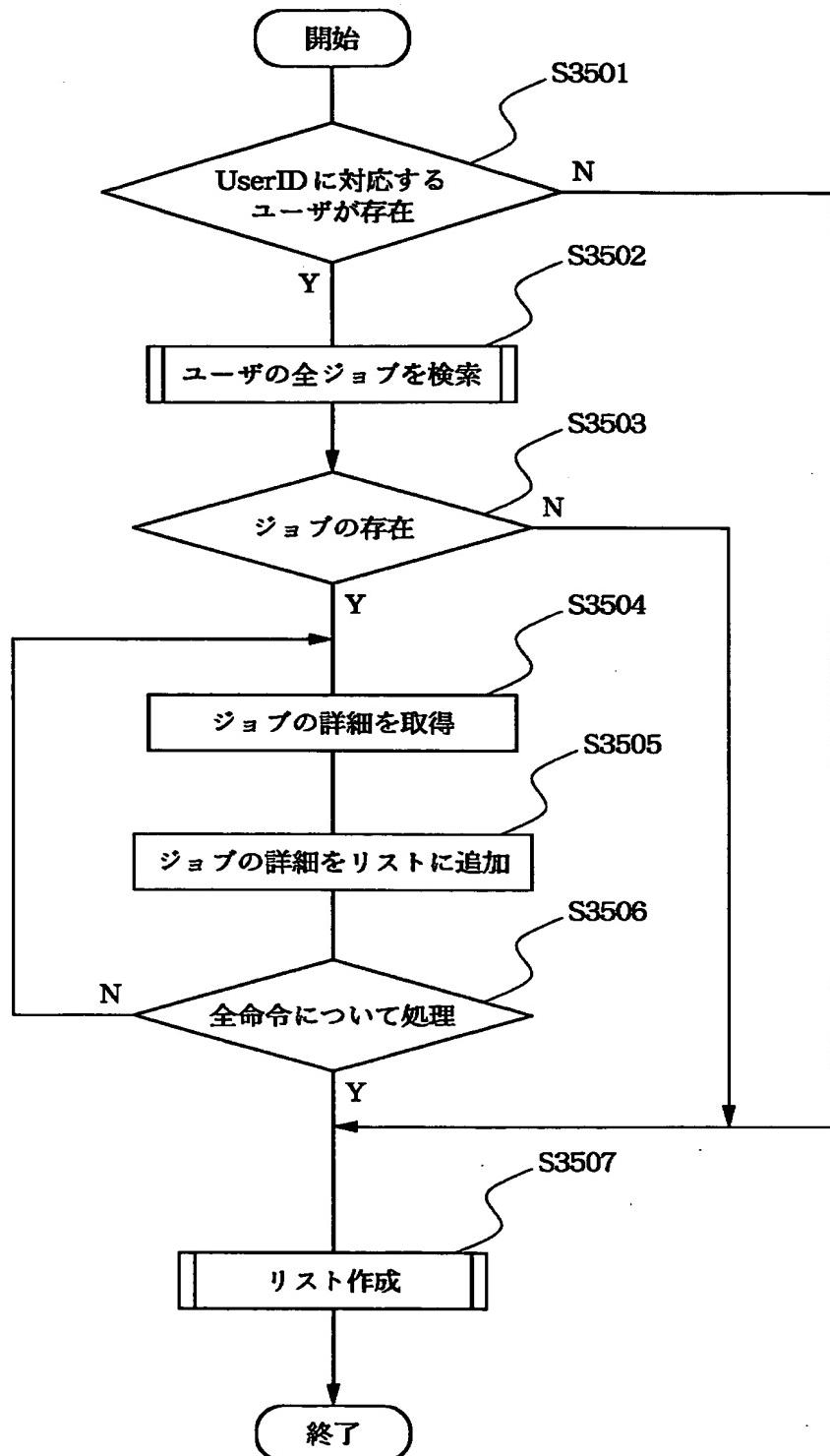
【図33】



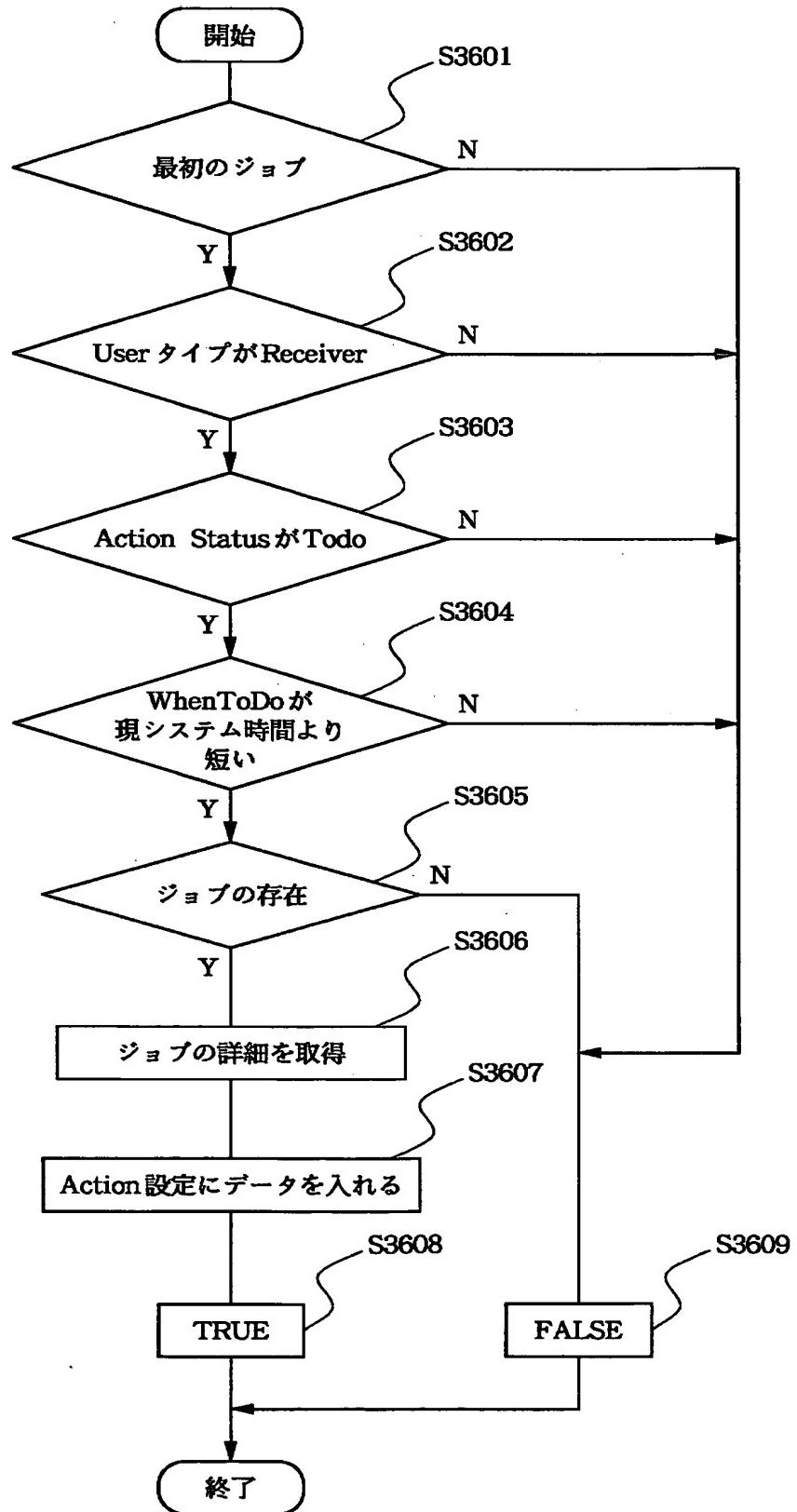
【図34】



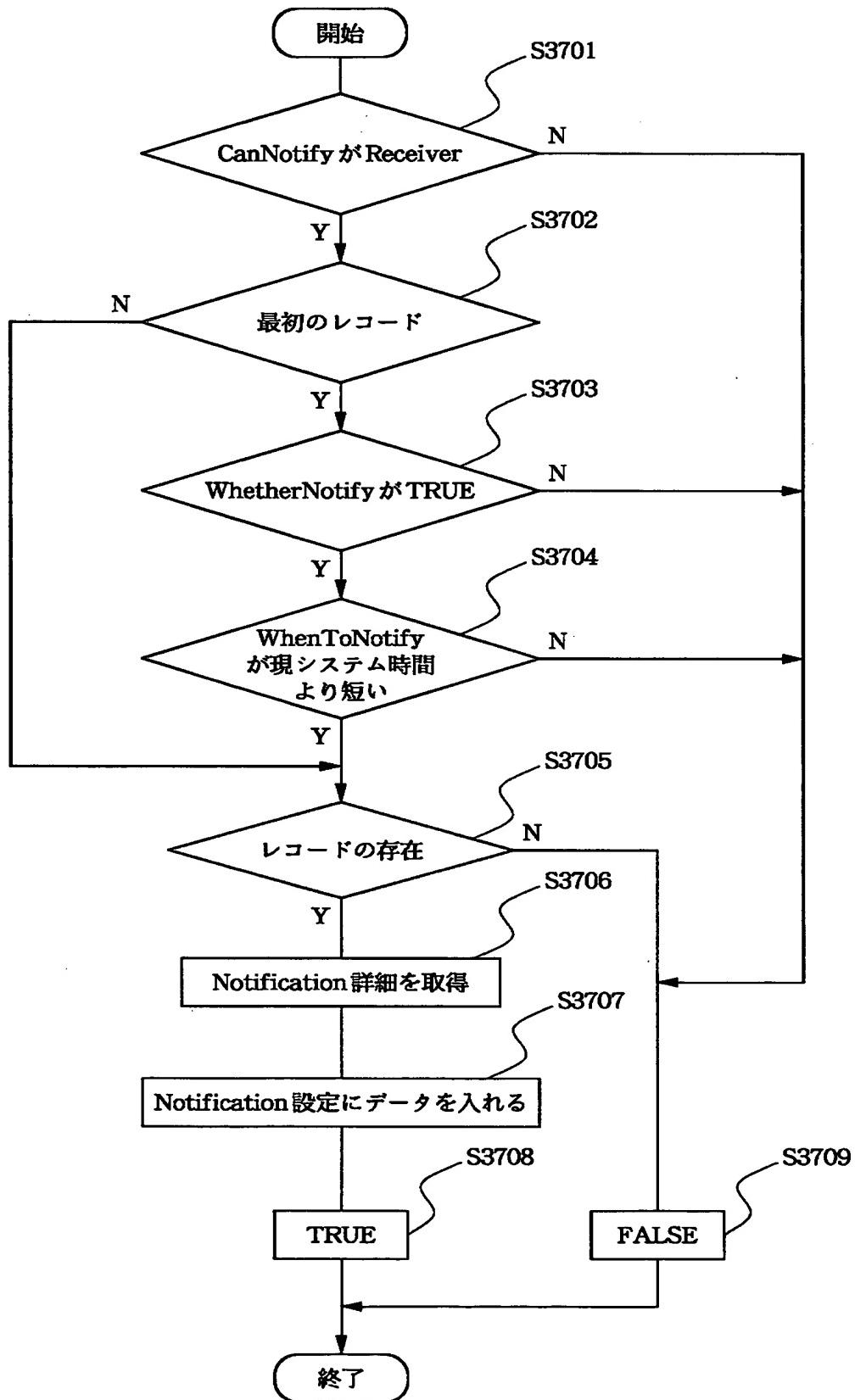
【図35】



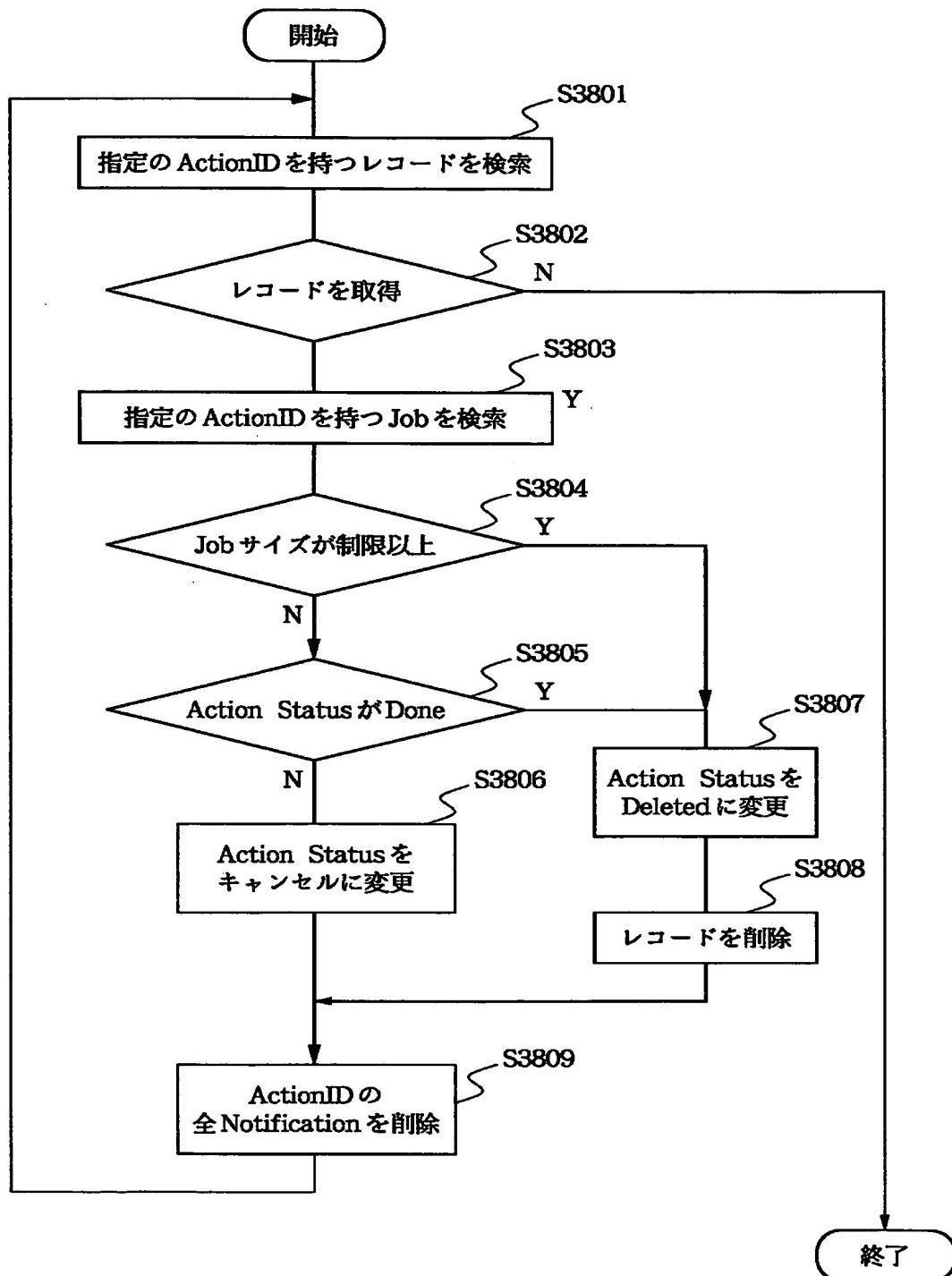
【図36】



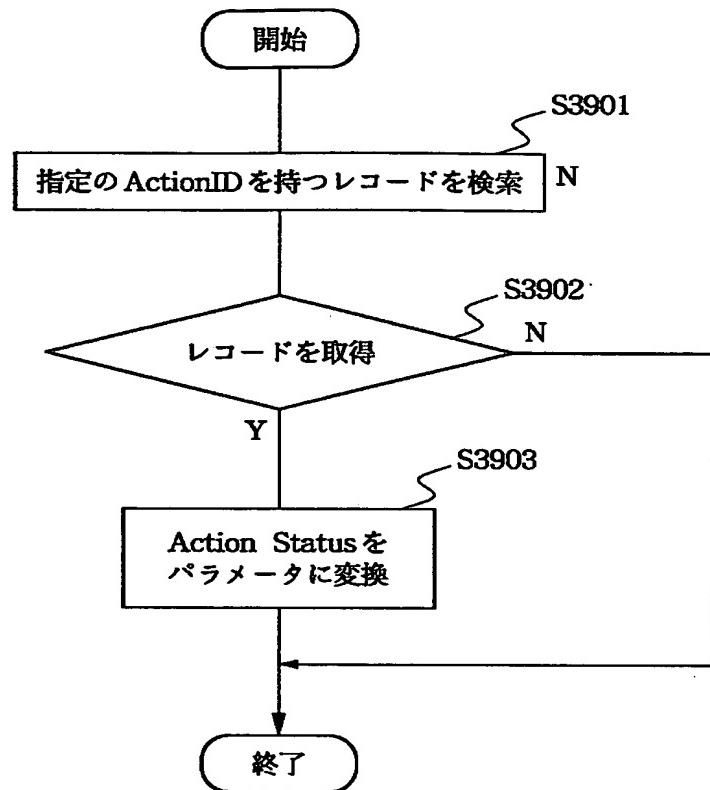
【図37】



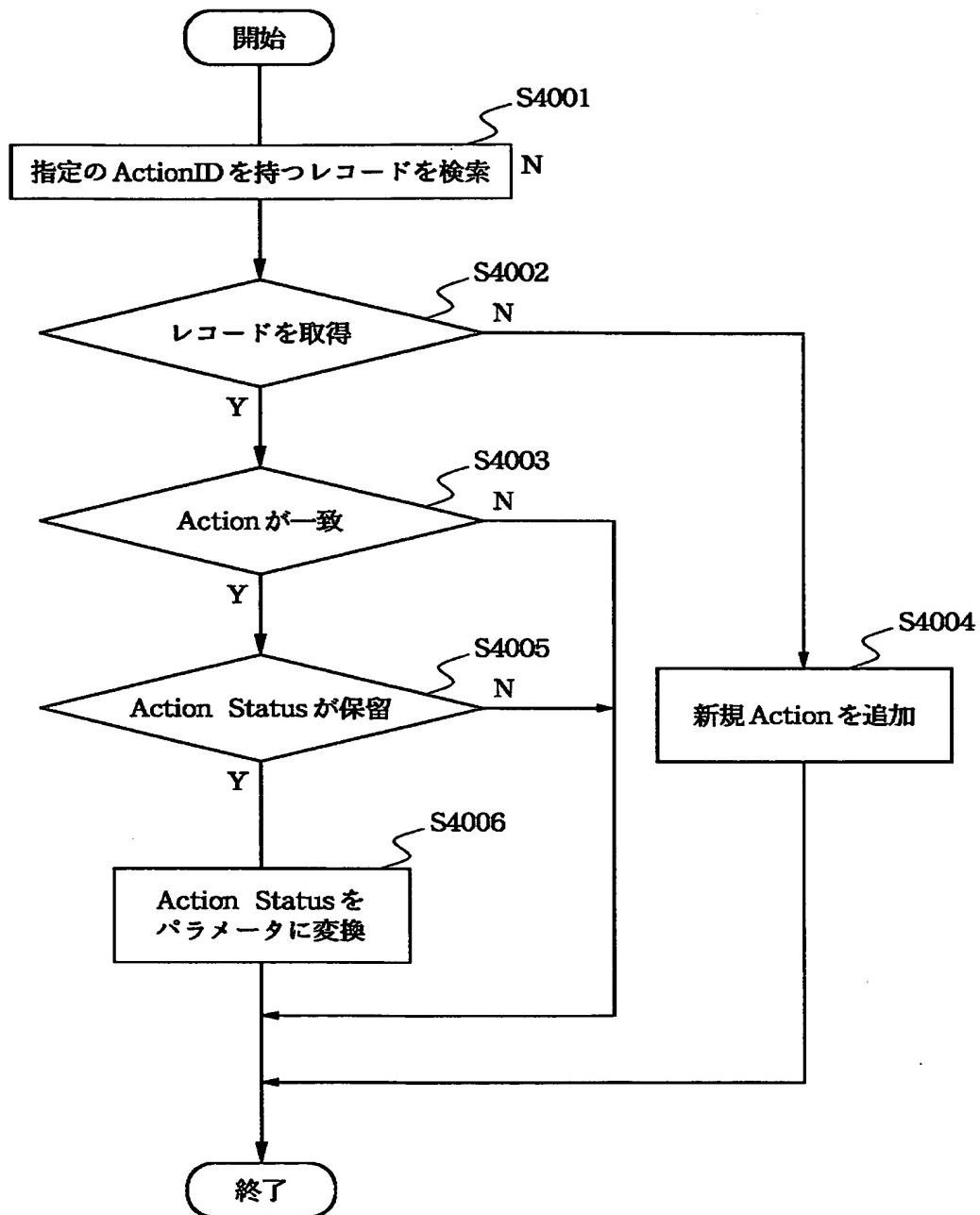
【図38】



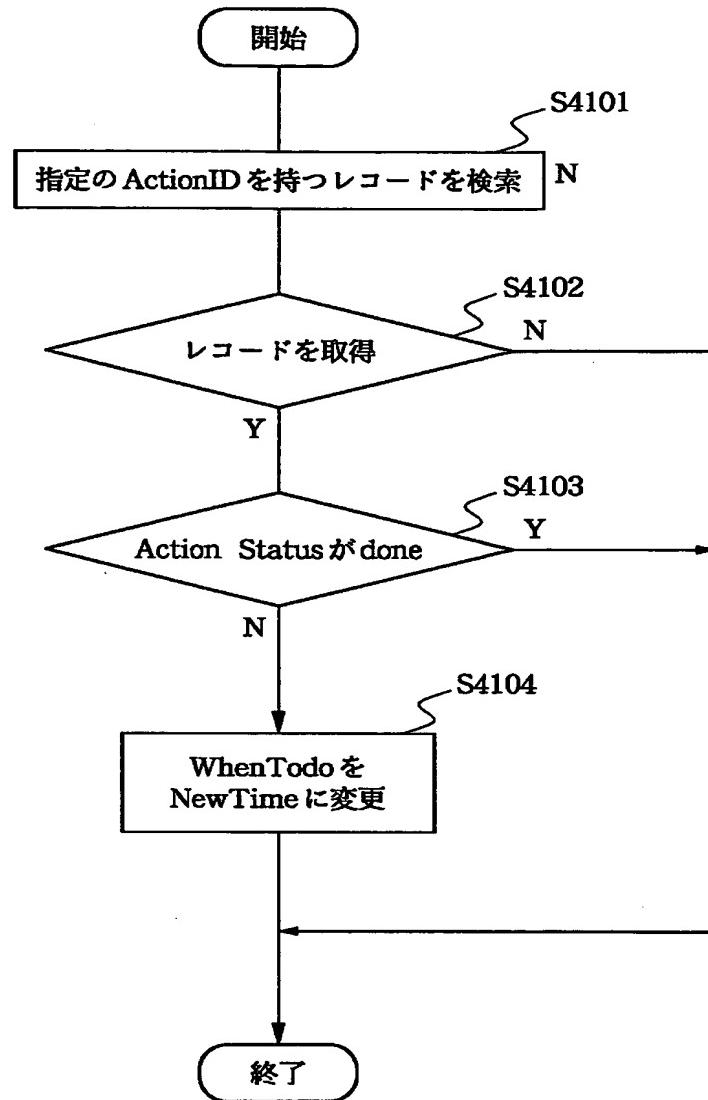
【図39】



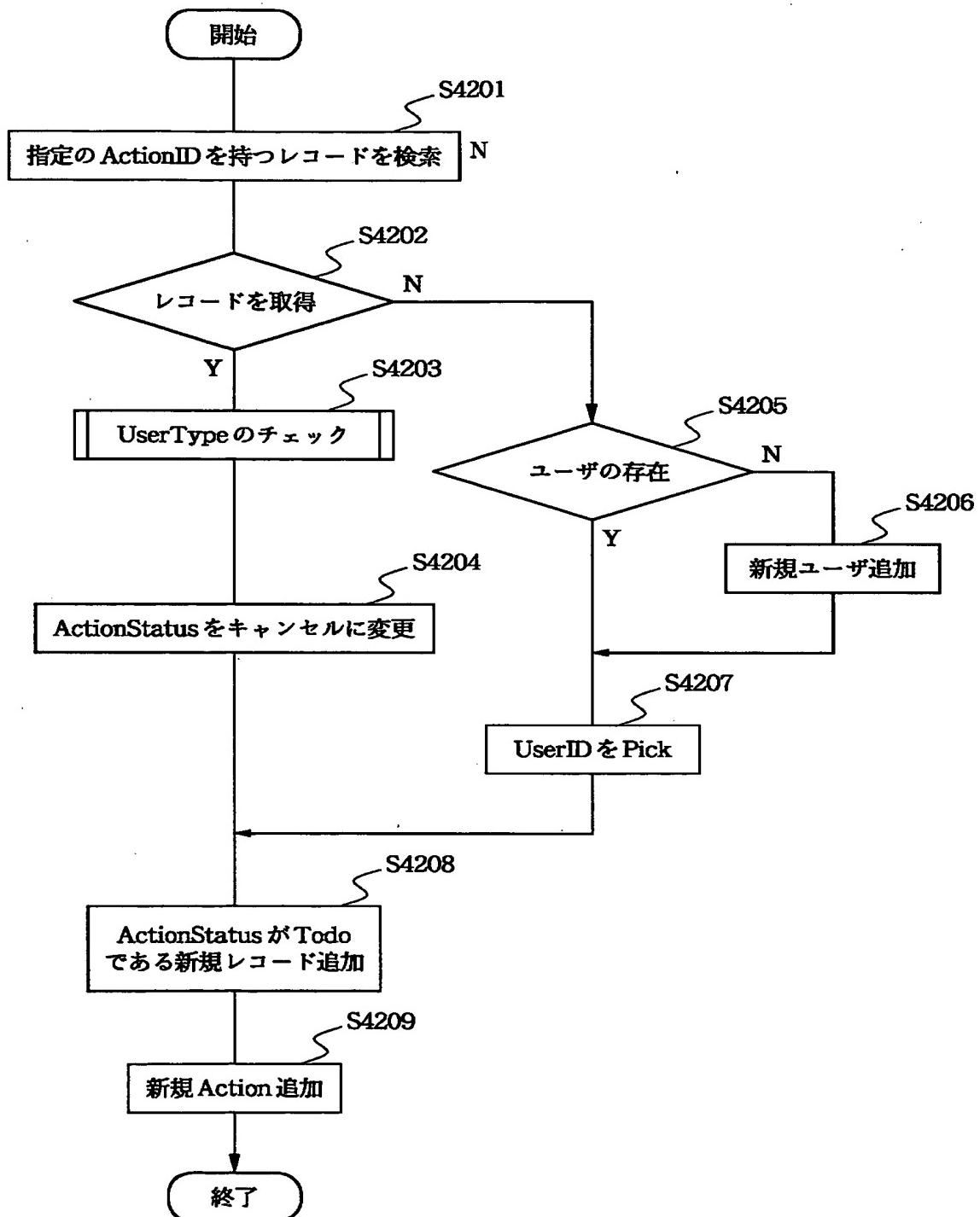
【図40】



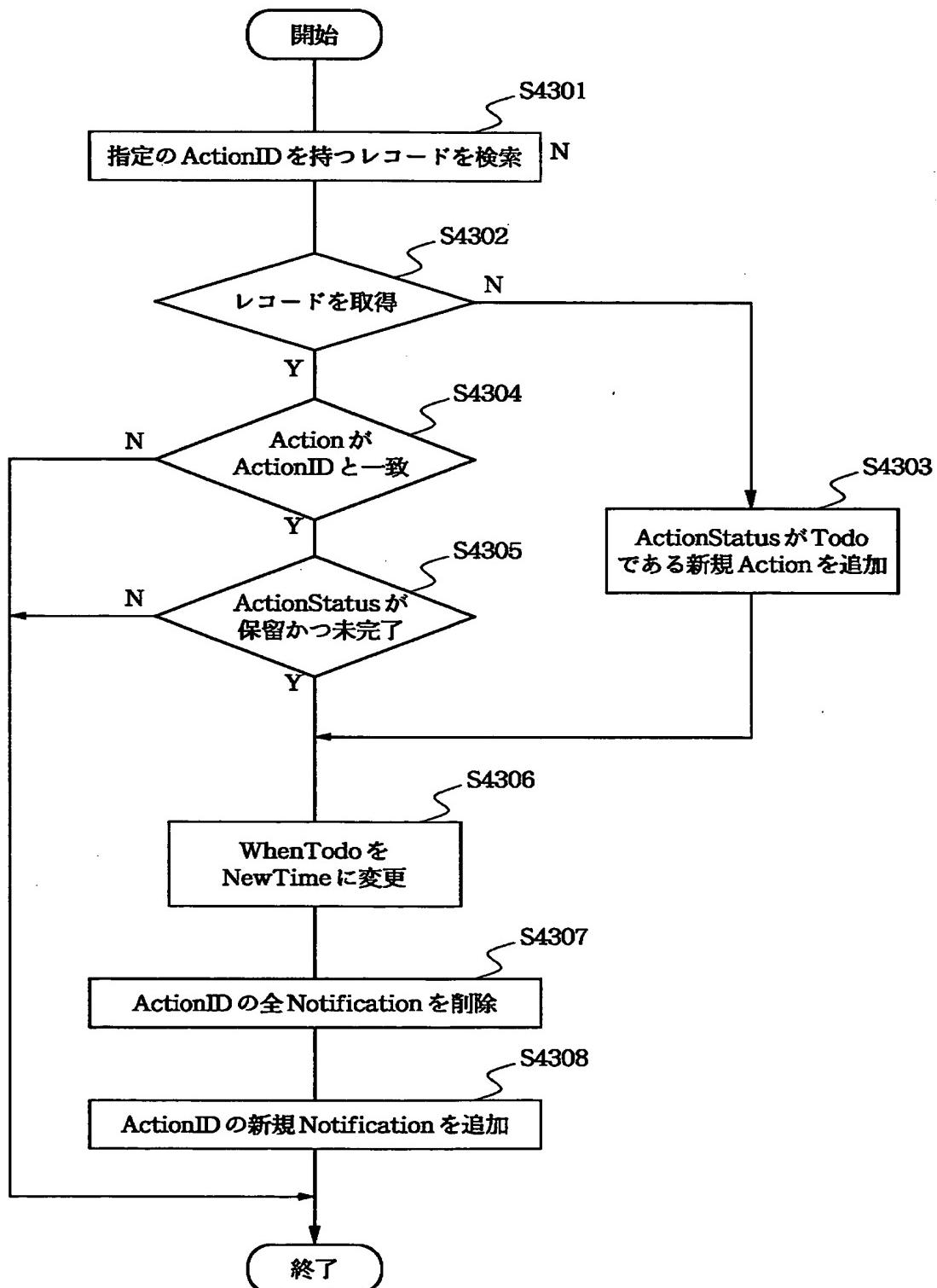
【図41】



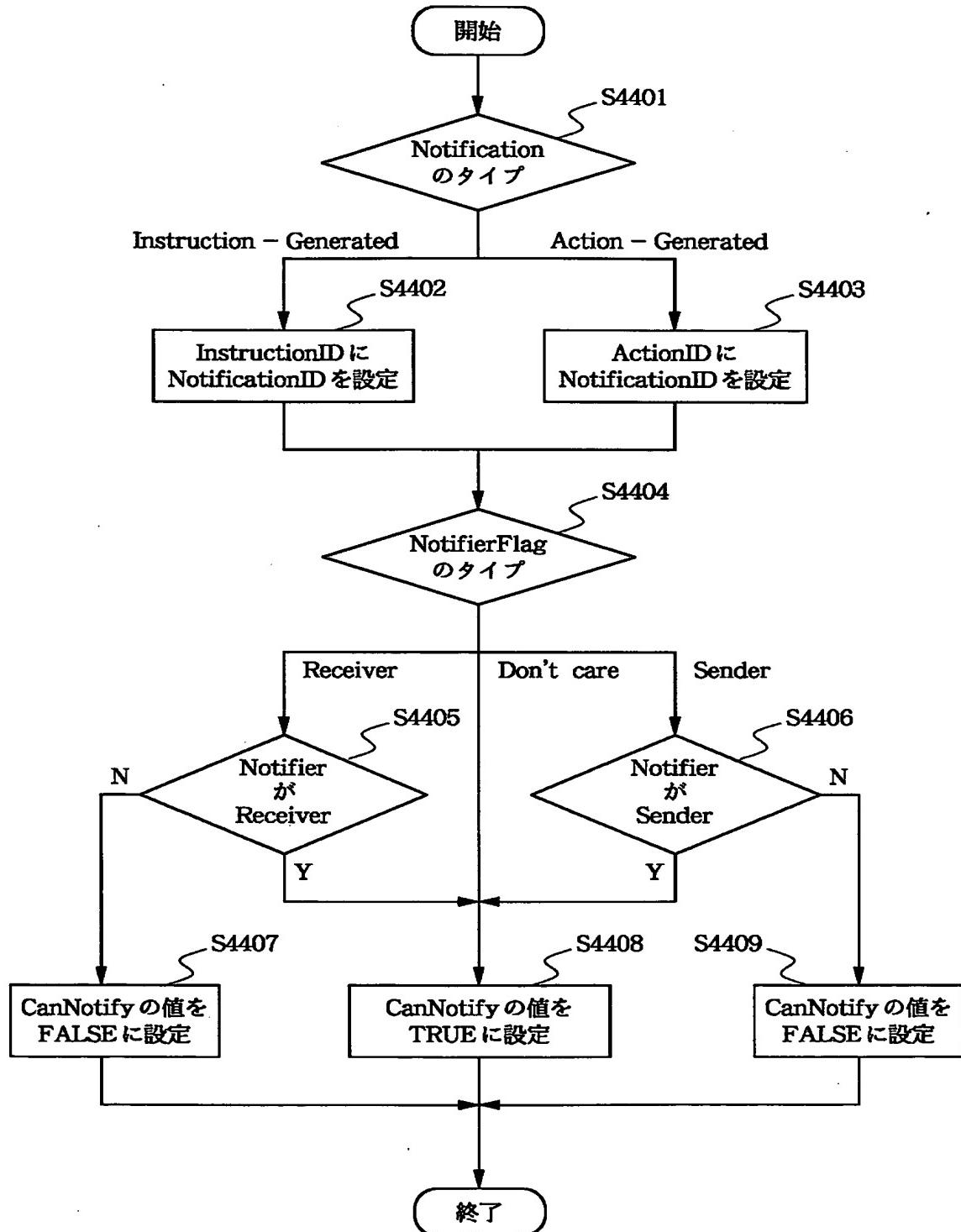
【図42】



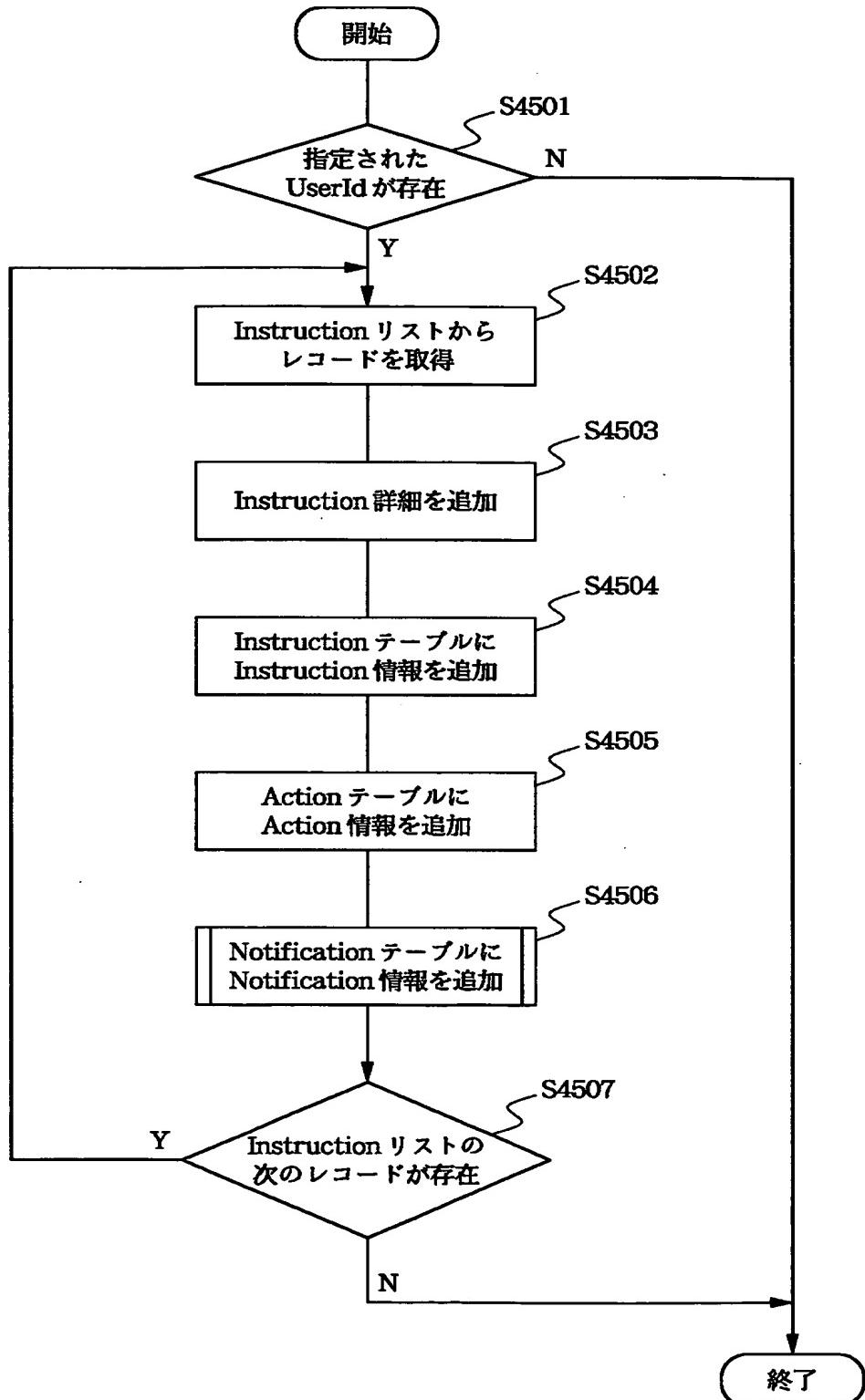
【図4-3】



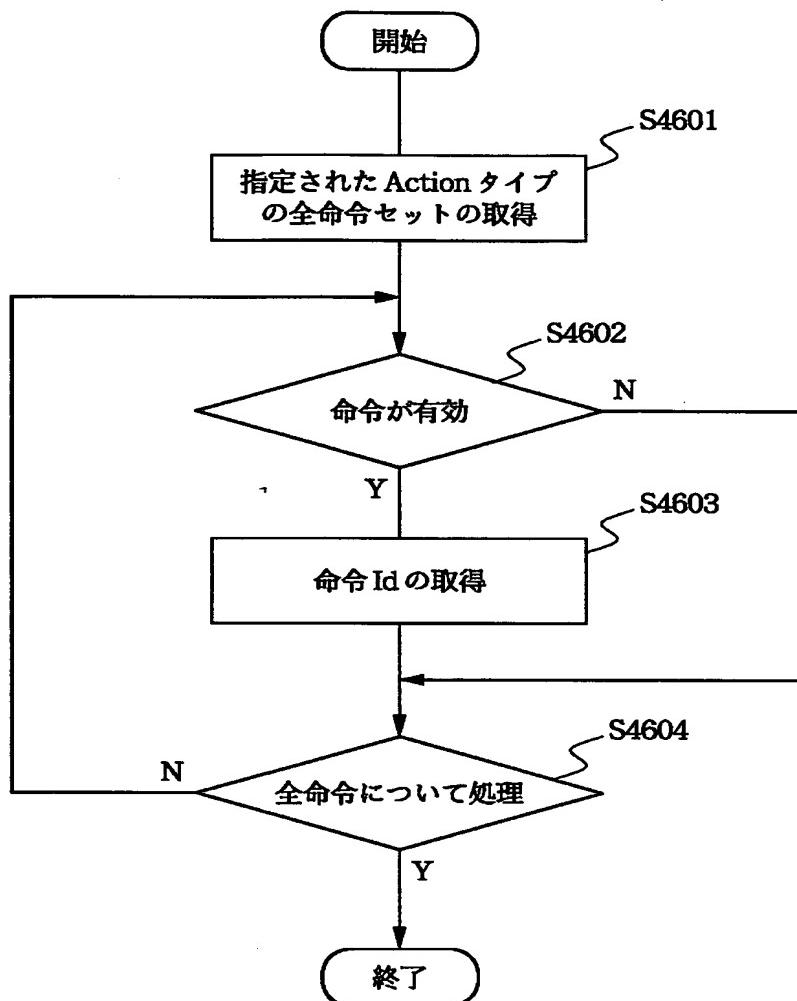
【図44】



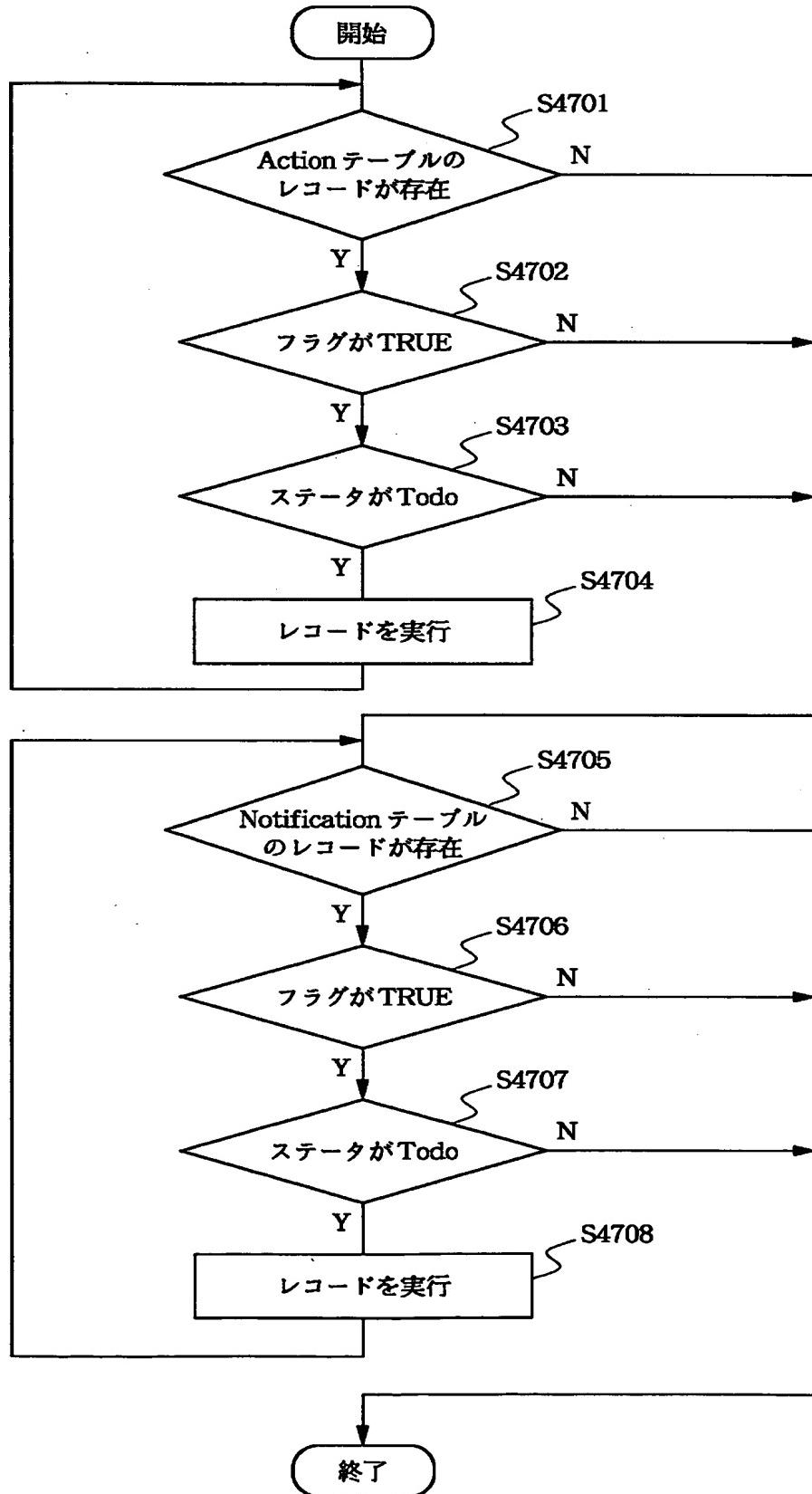
【図45】



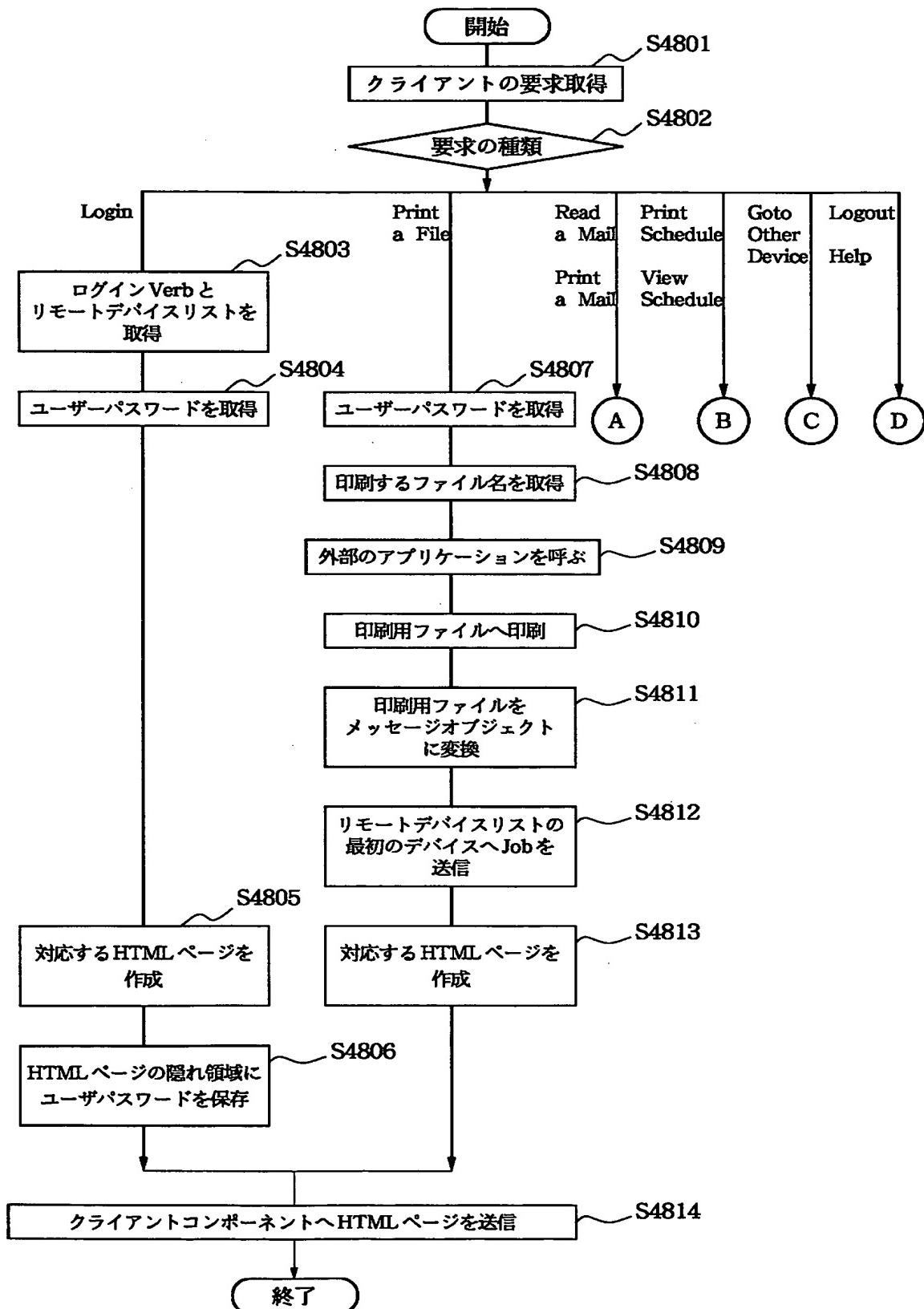
【図46】



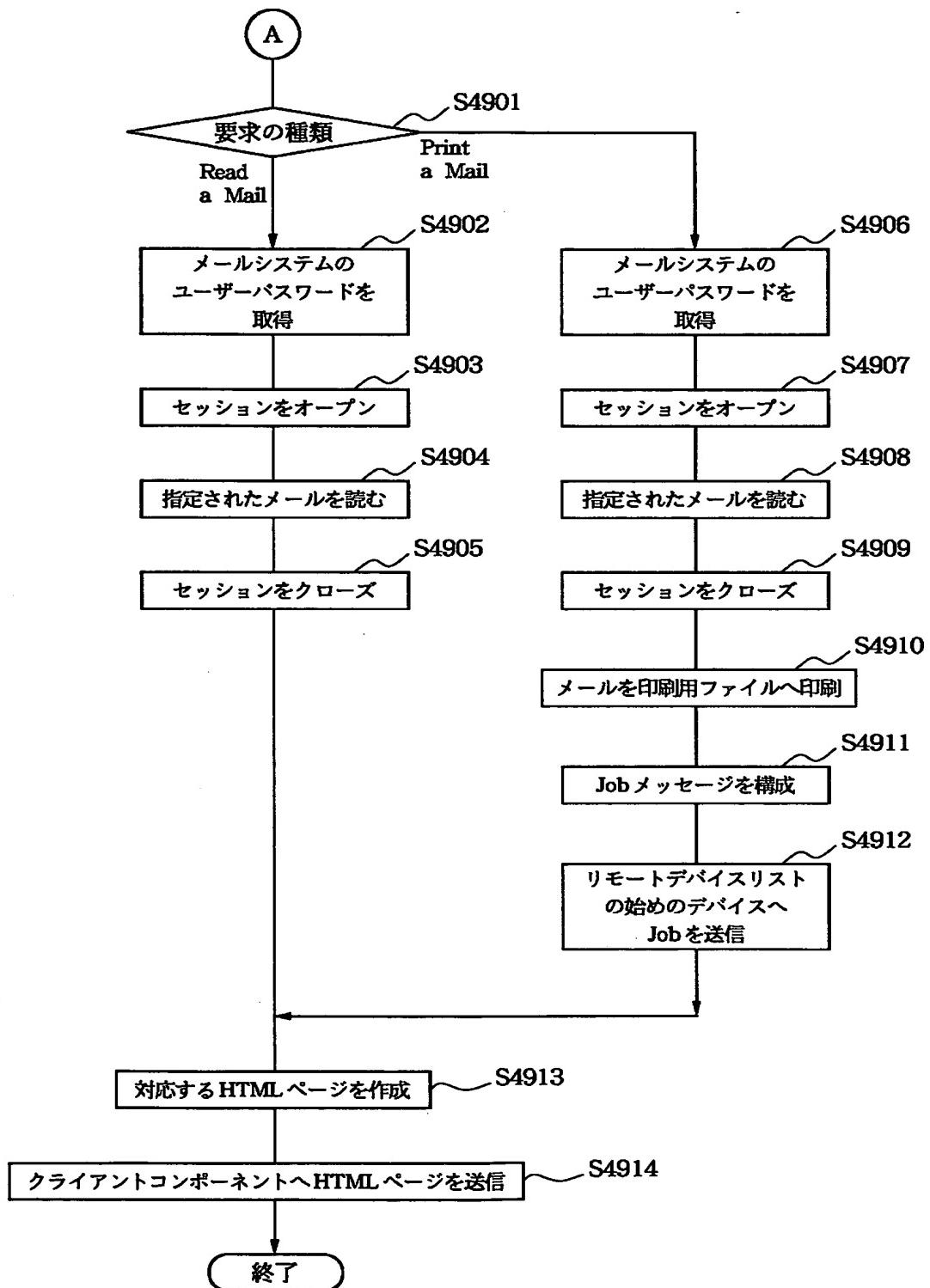
【図47】



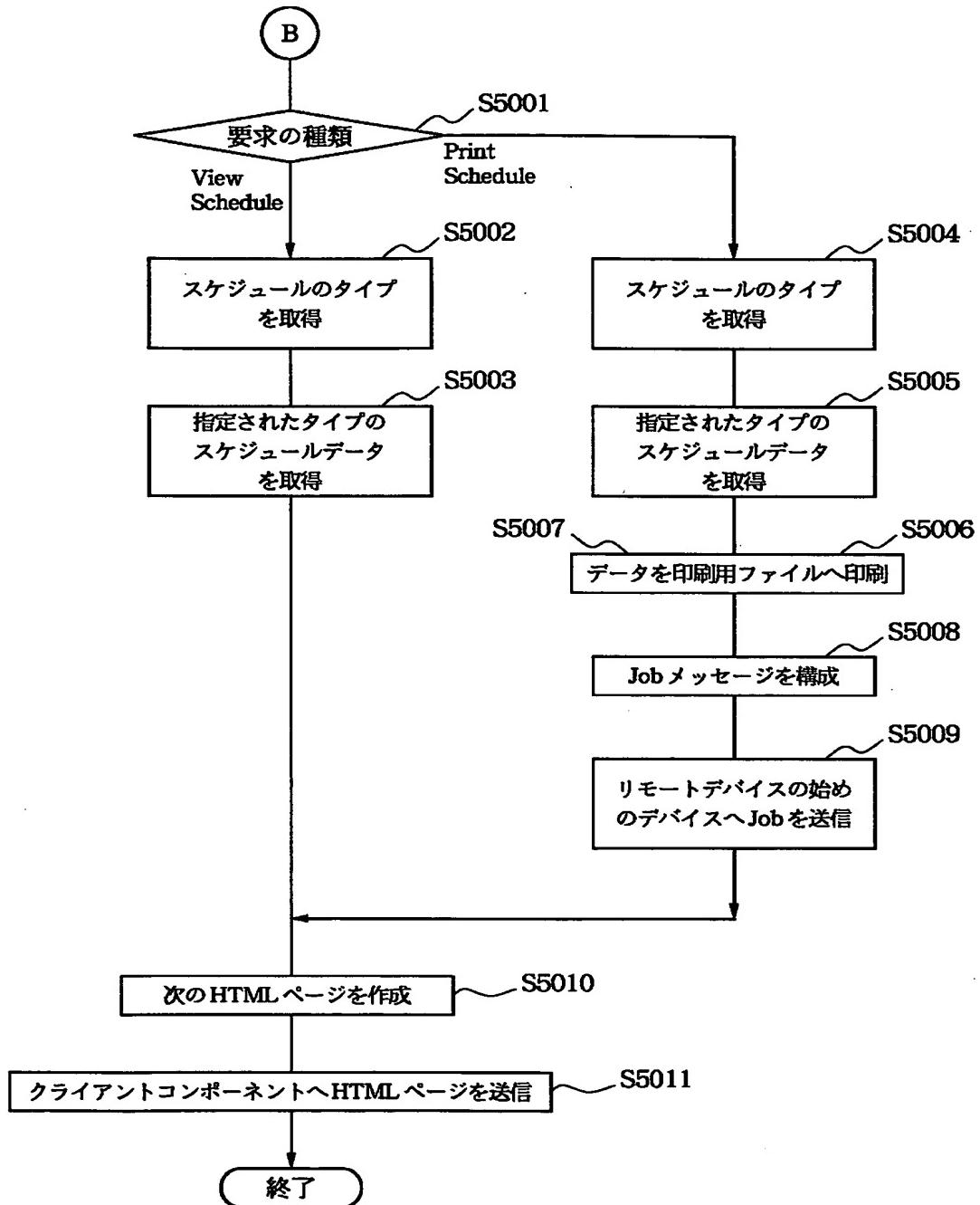
【図4.8】



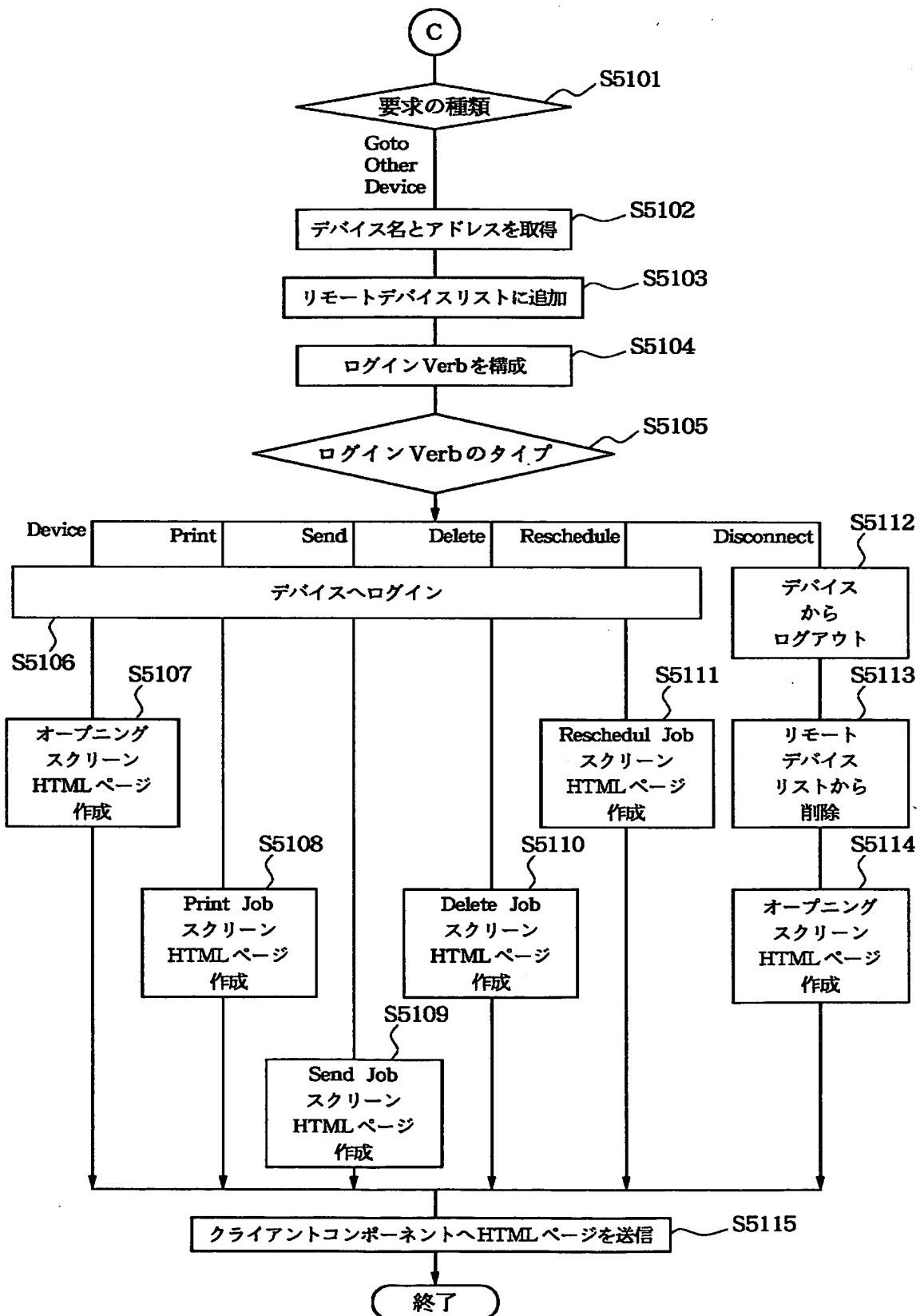
【図49】



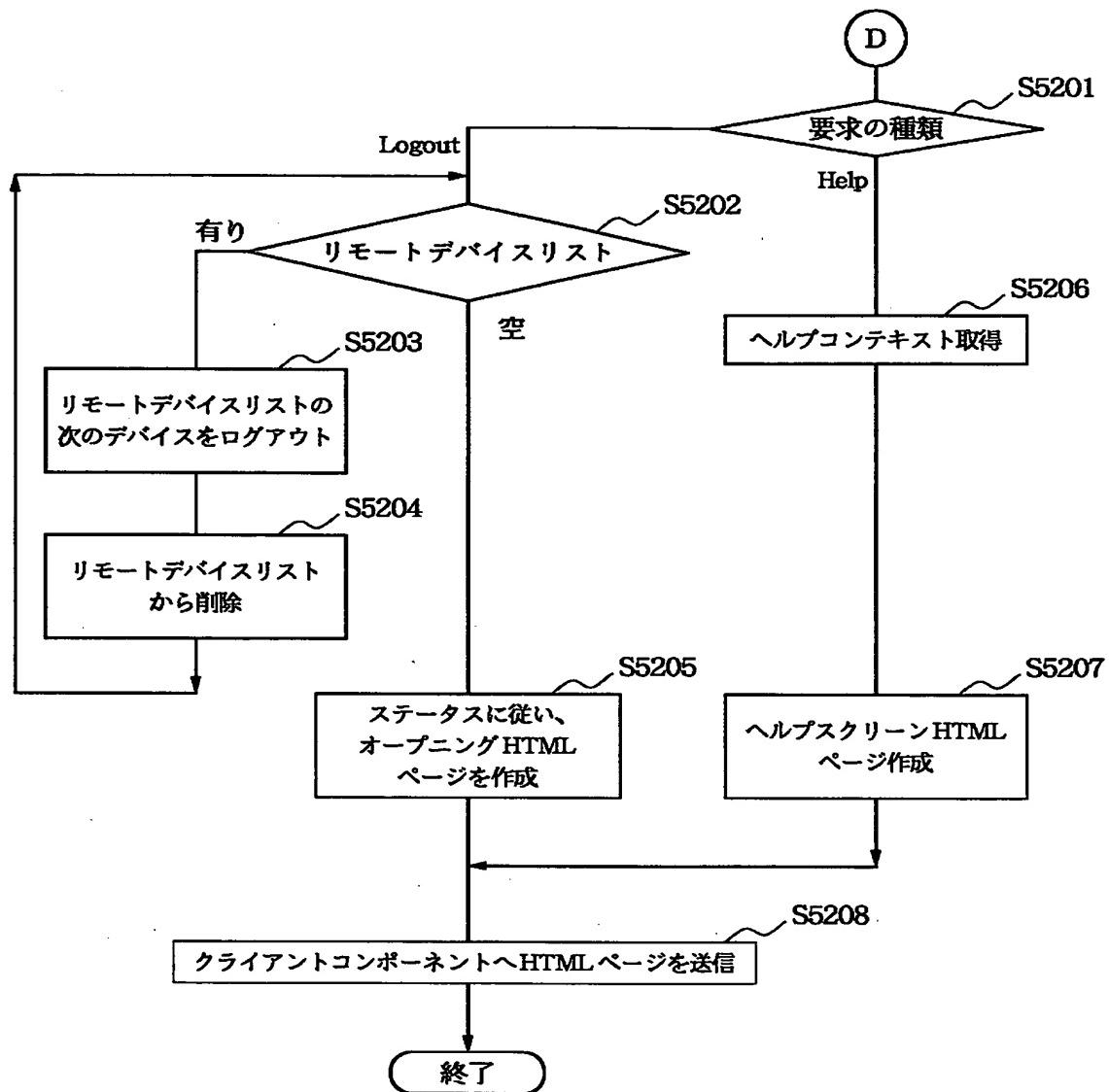
【図50】



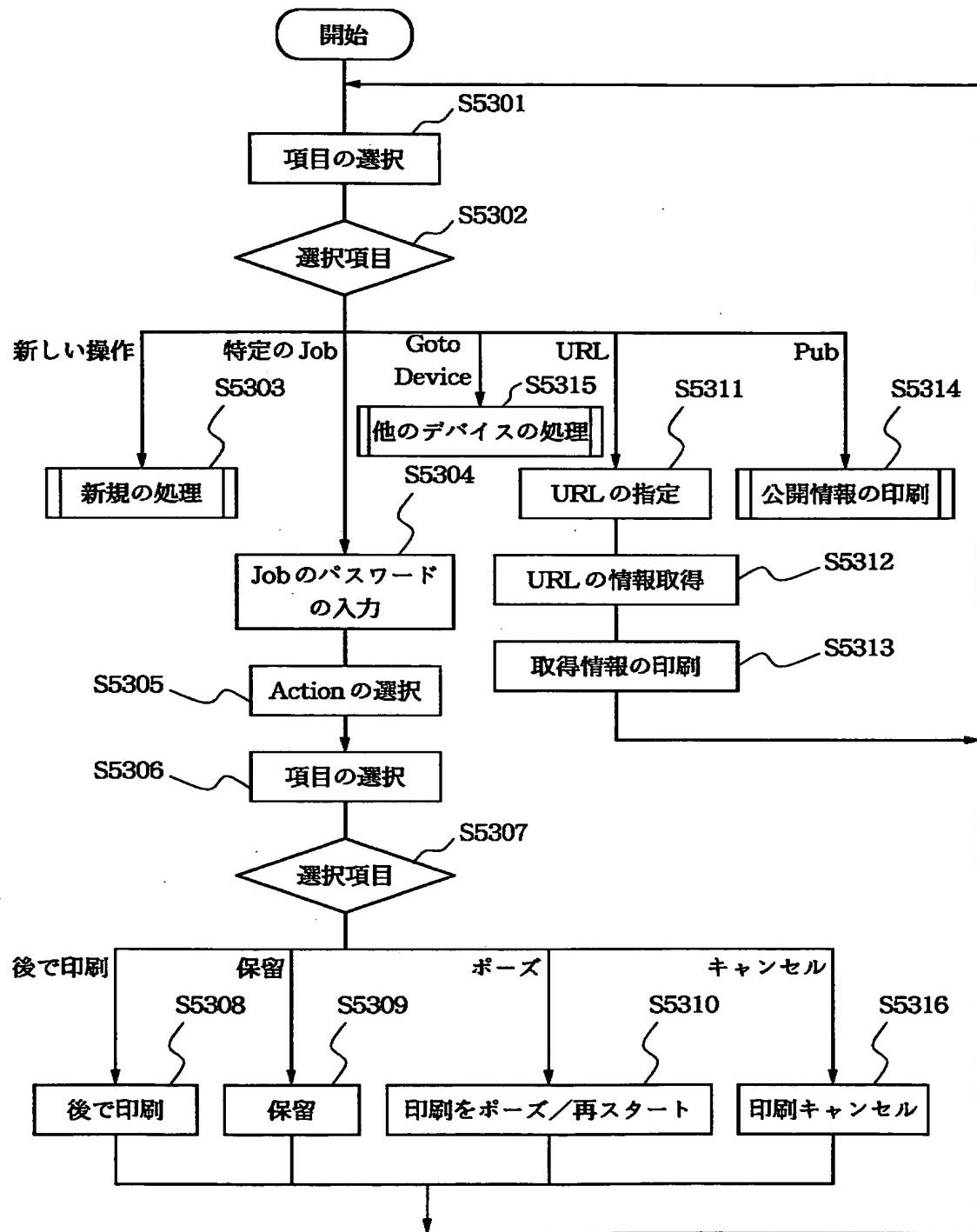
【図51】



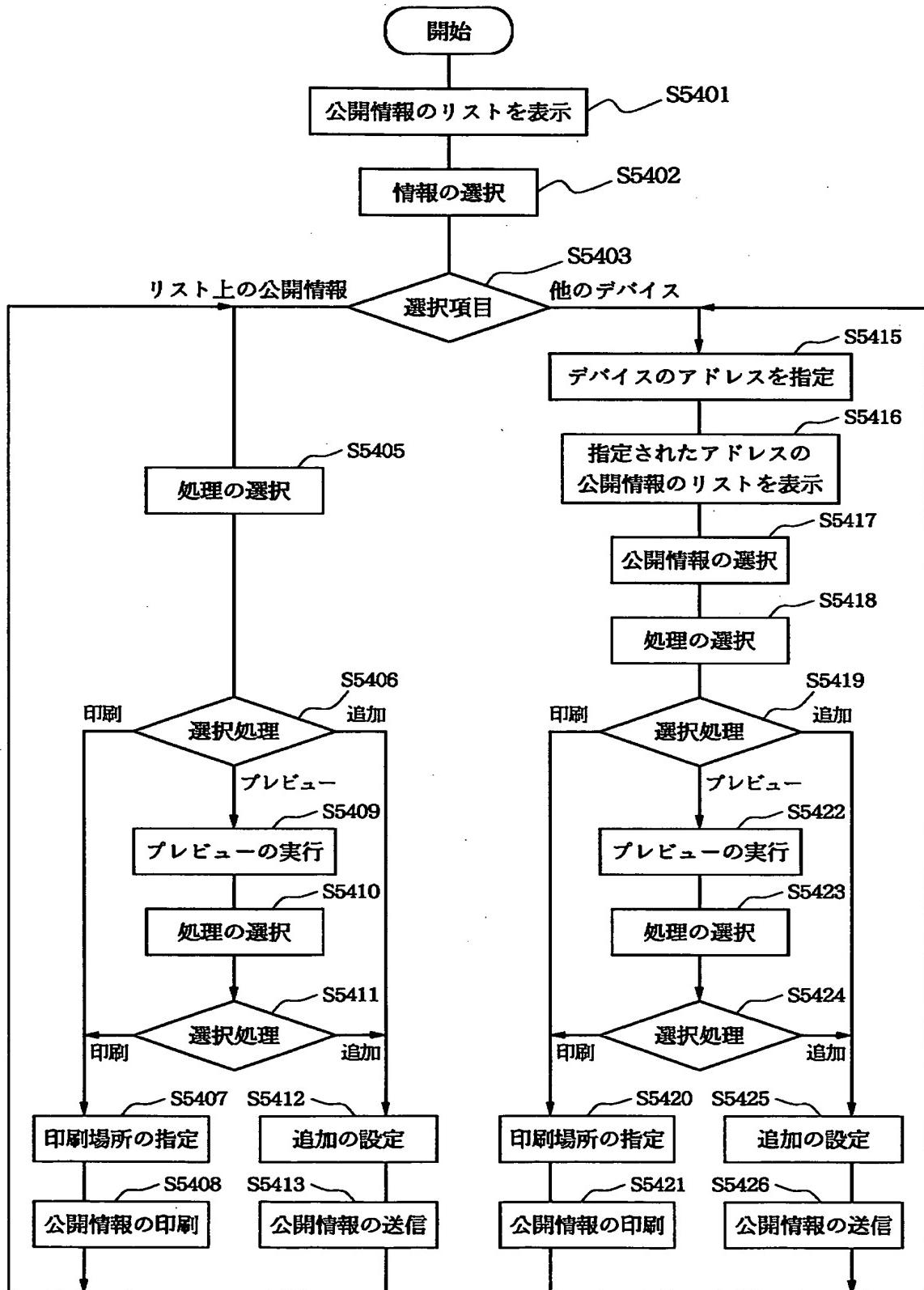
【図52】



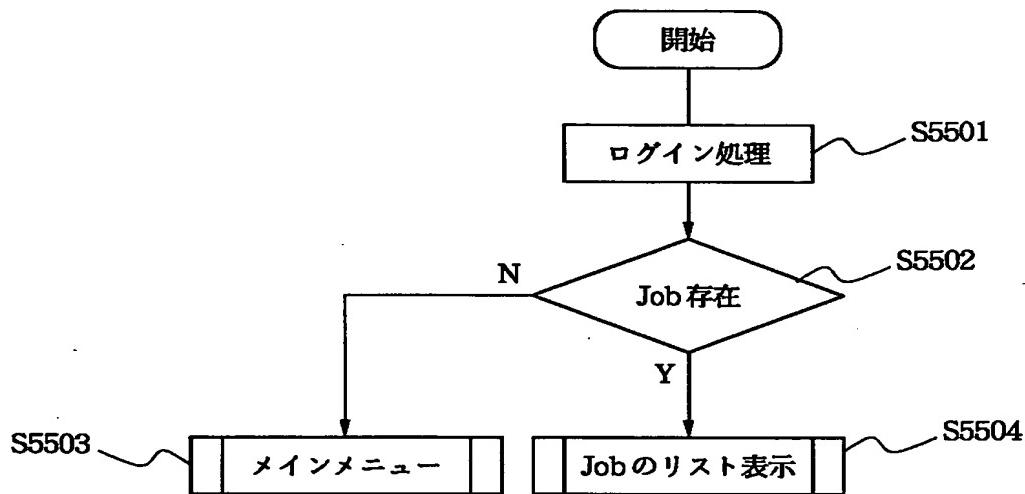
【図53】



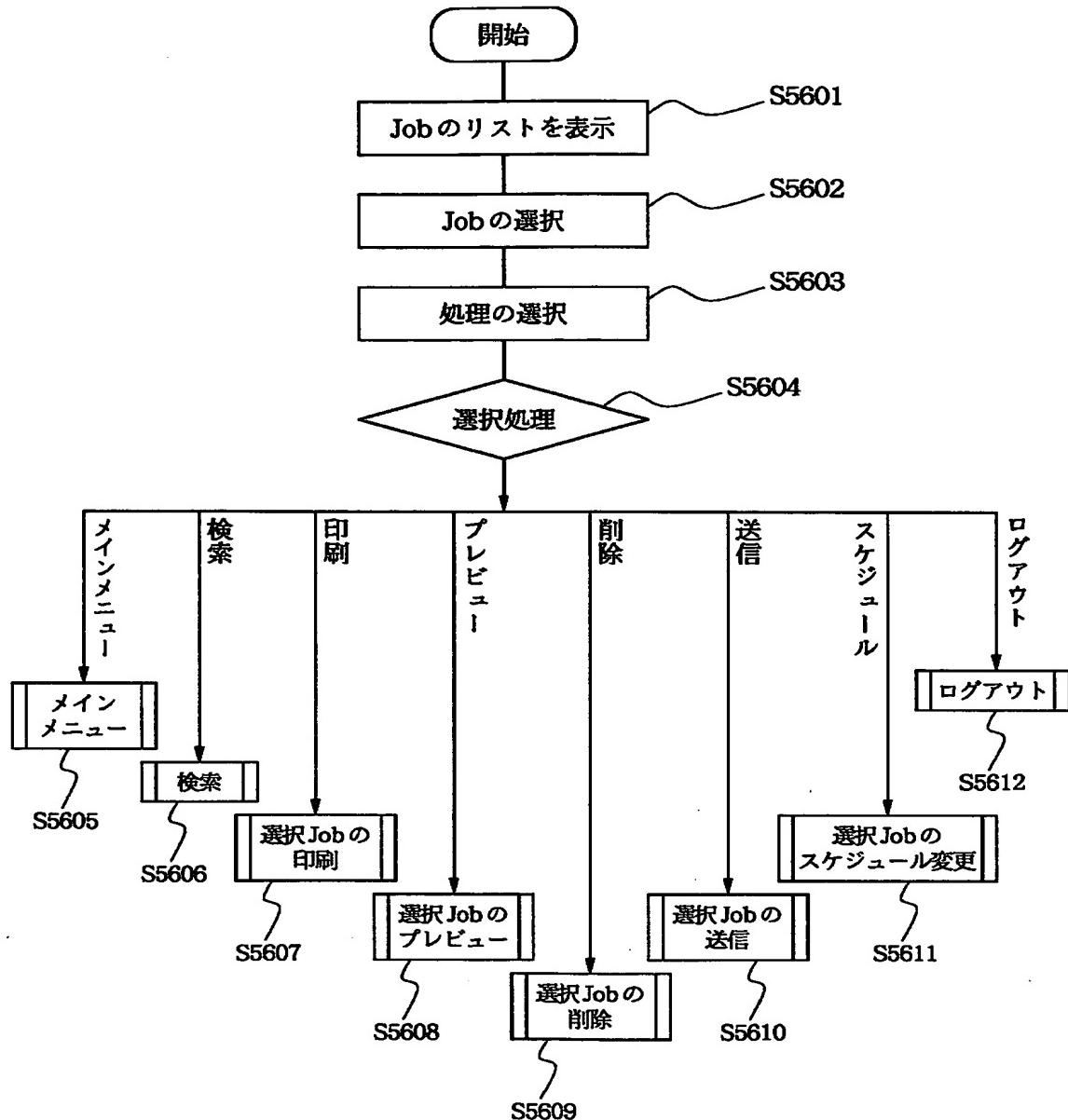
【図54】



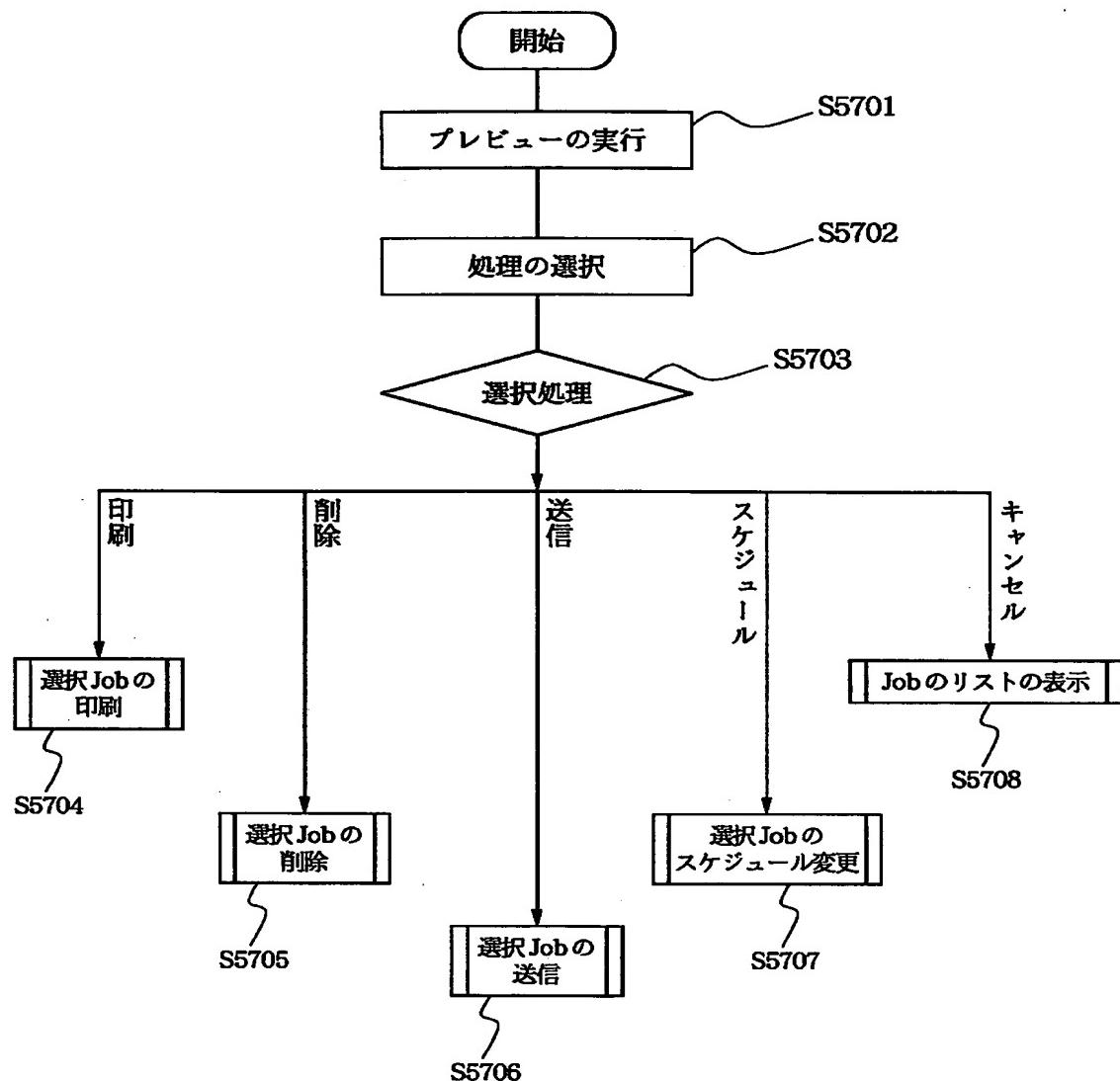
【図55】



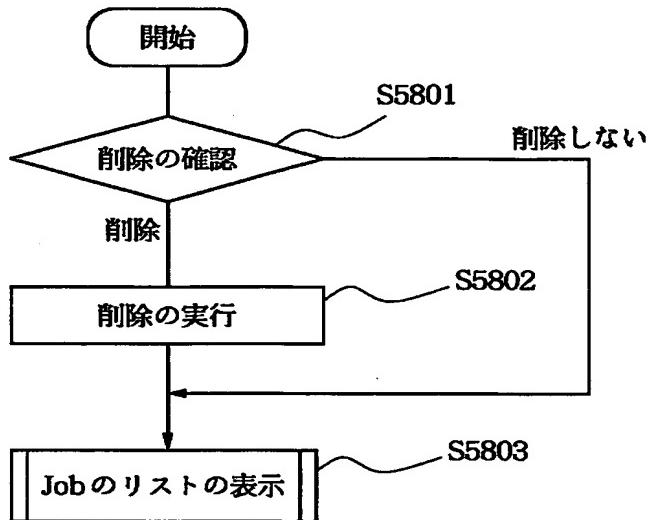
【図56】



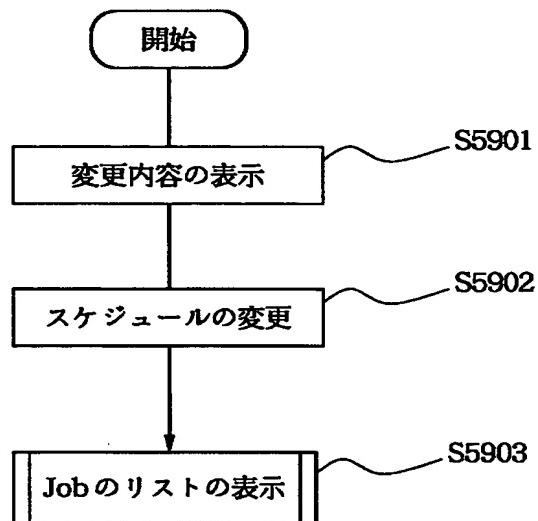
【図57】



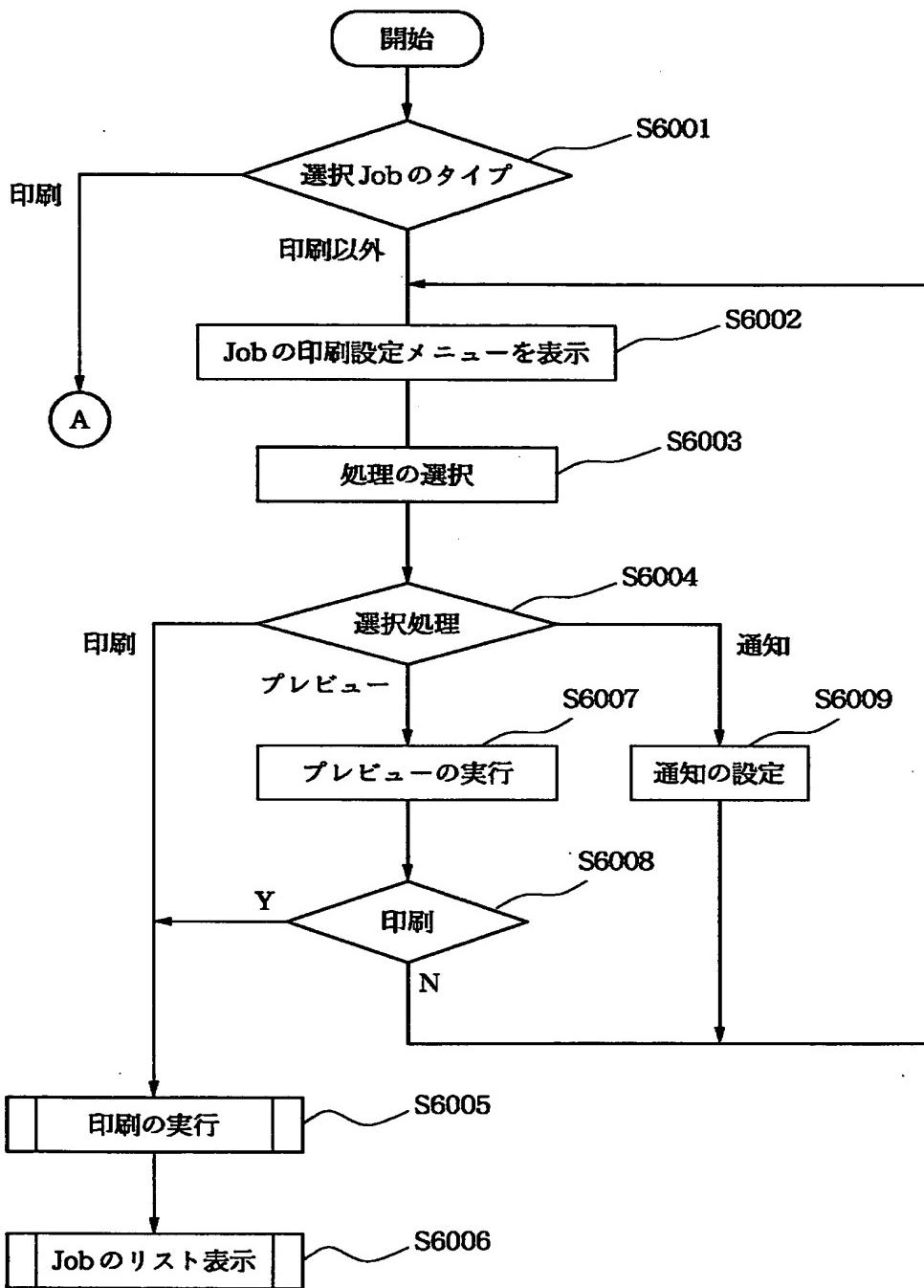
【図58】



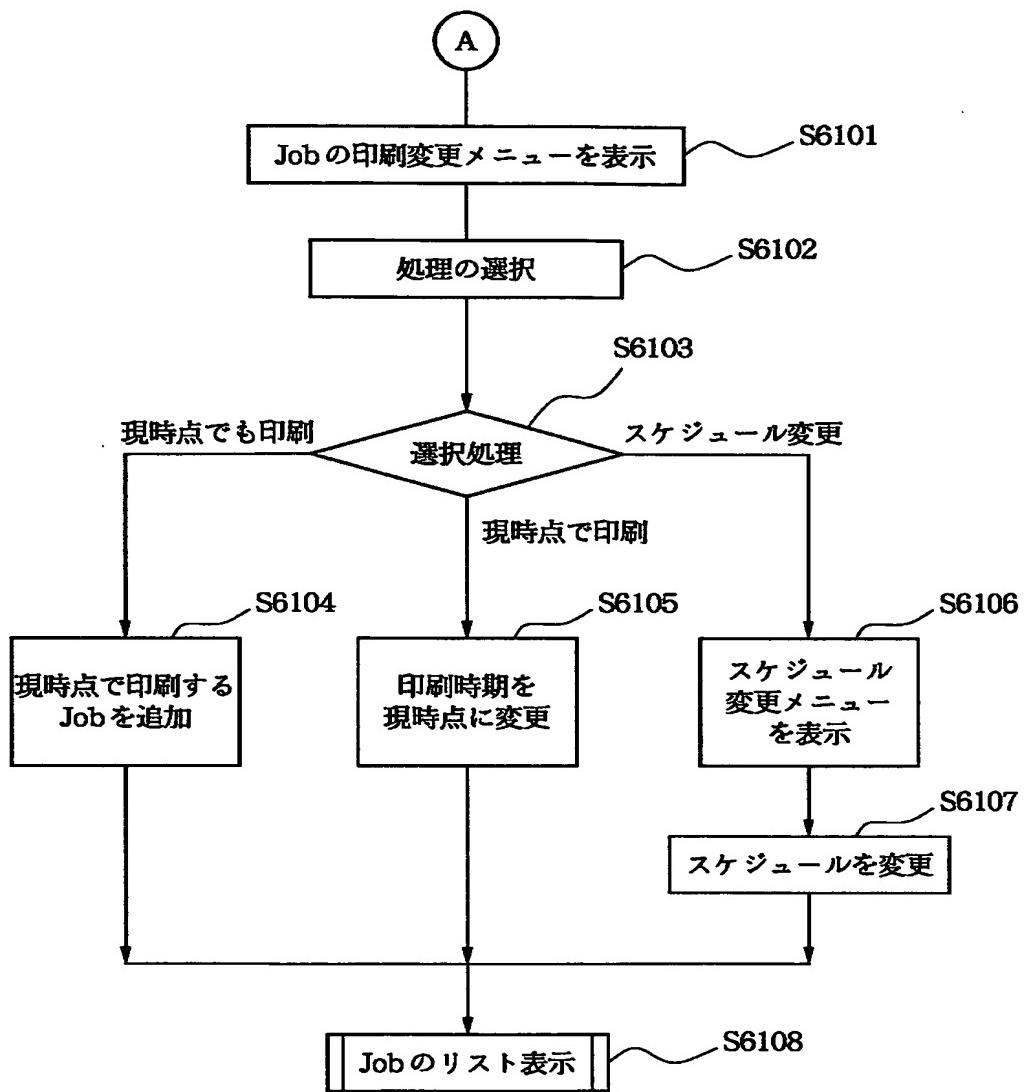
【図59】



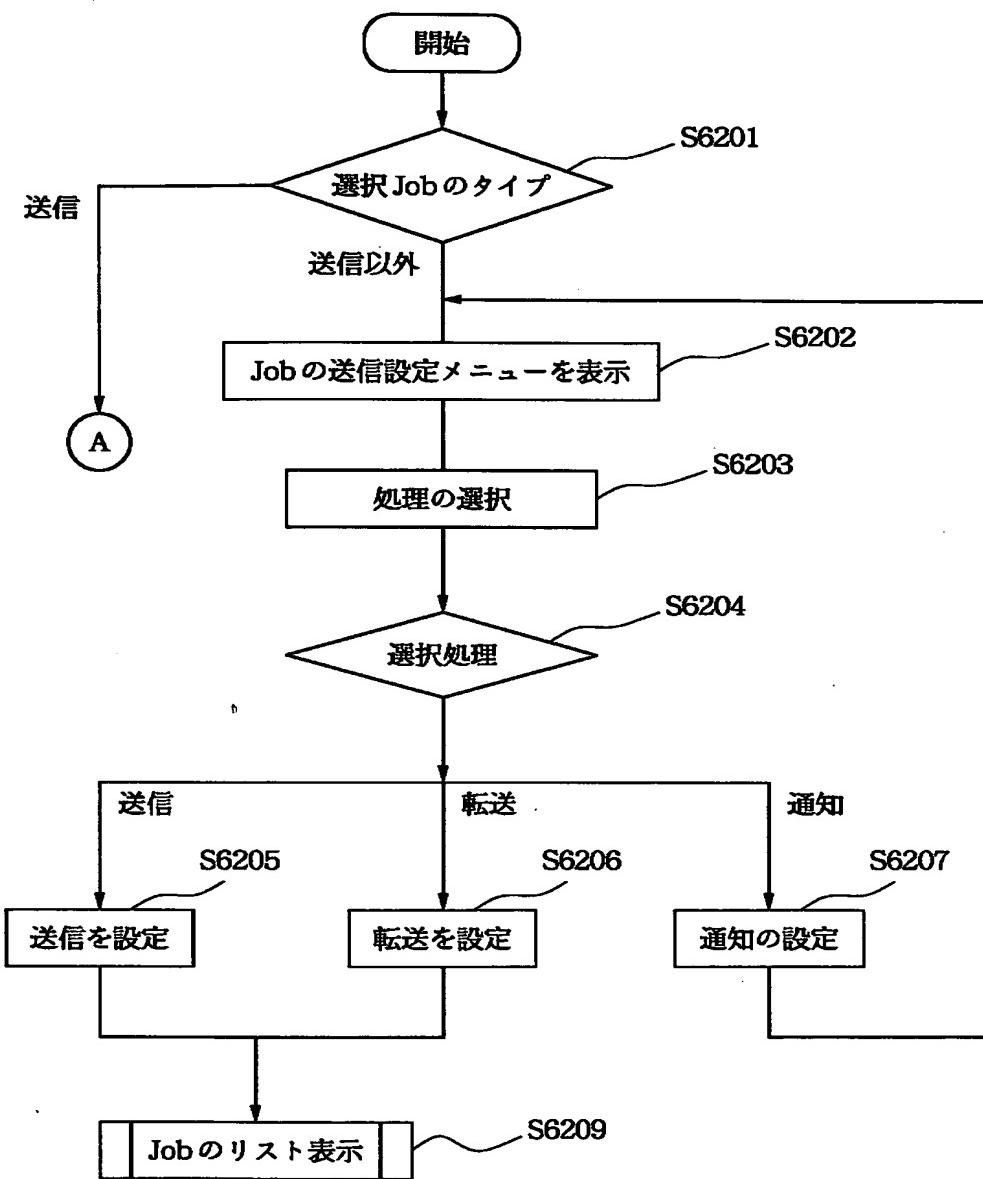
【図60】



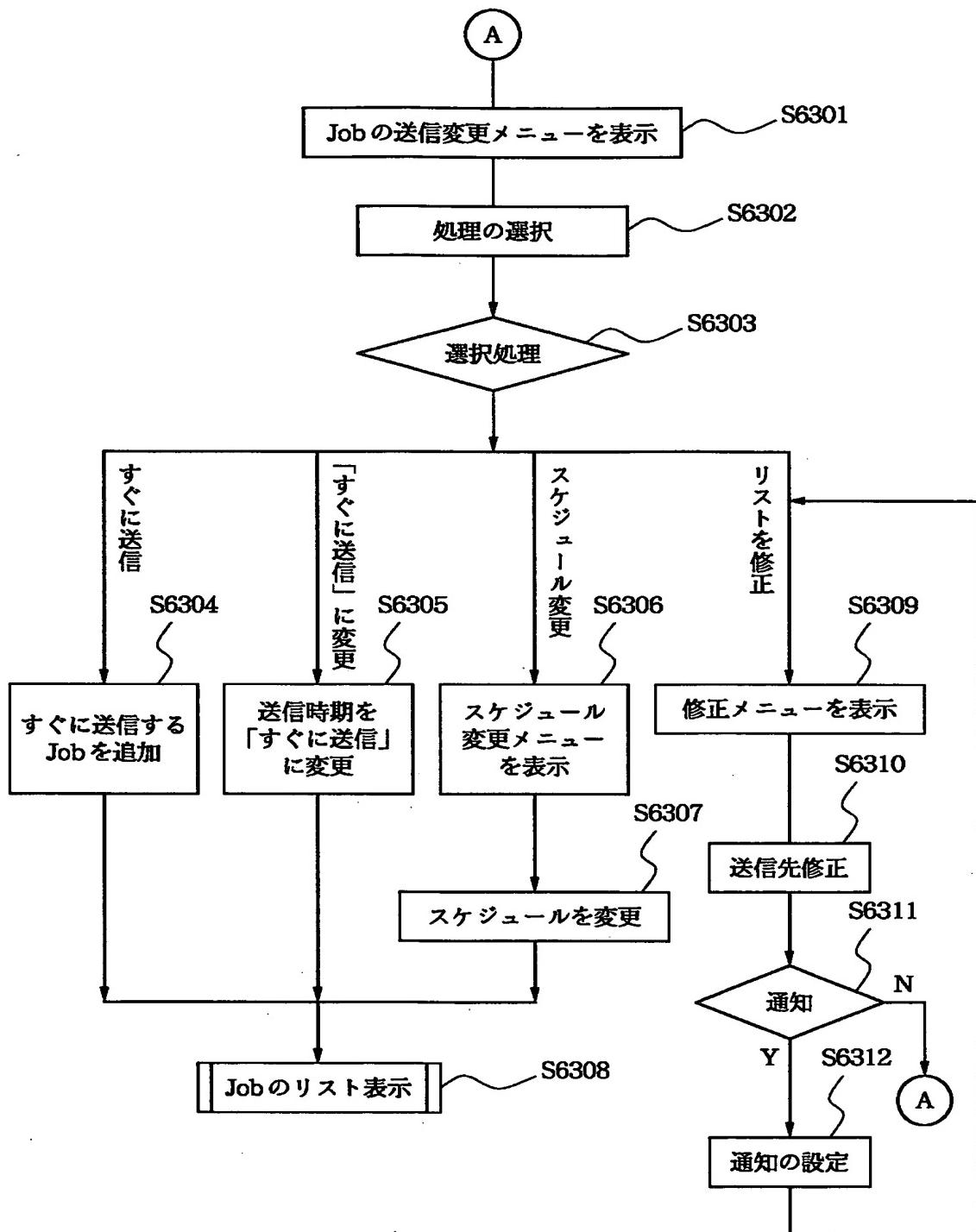
【図61】



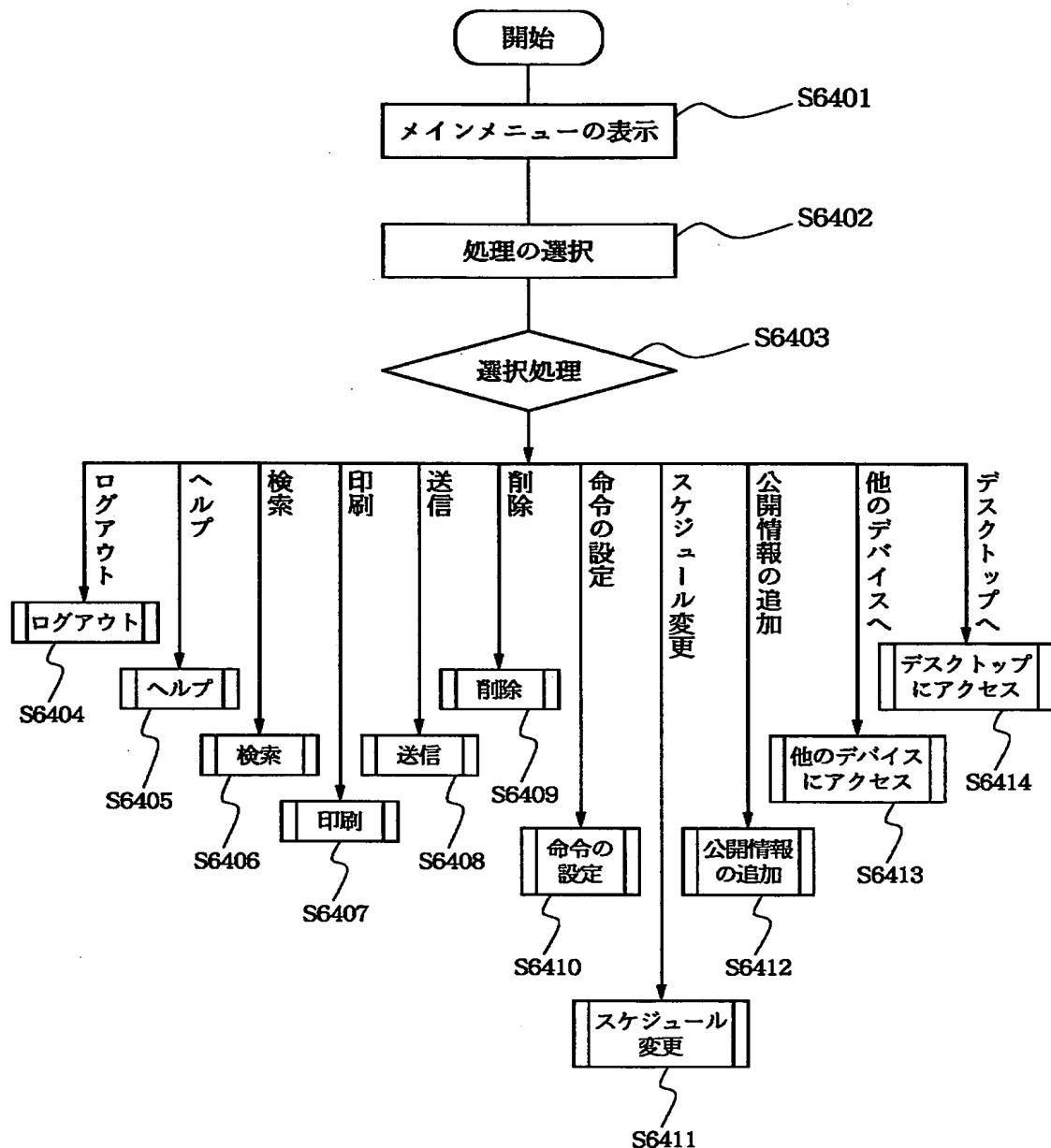
【図62】



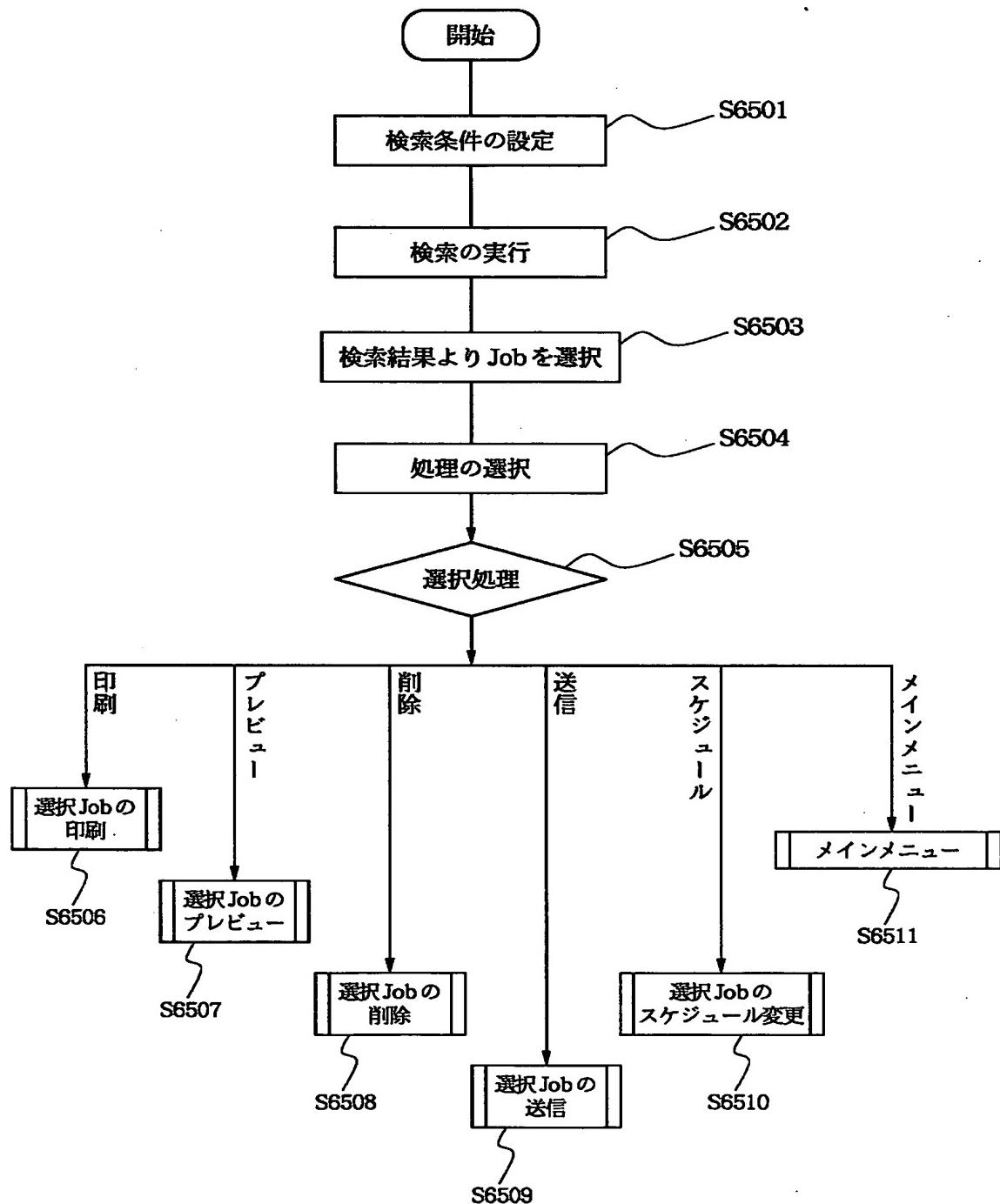
【図63】



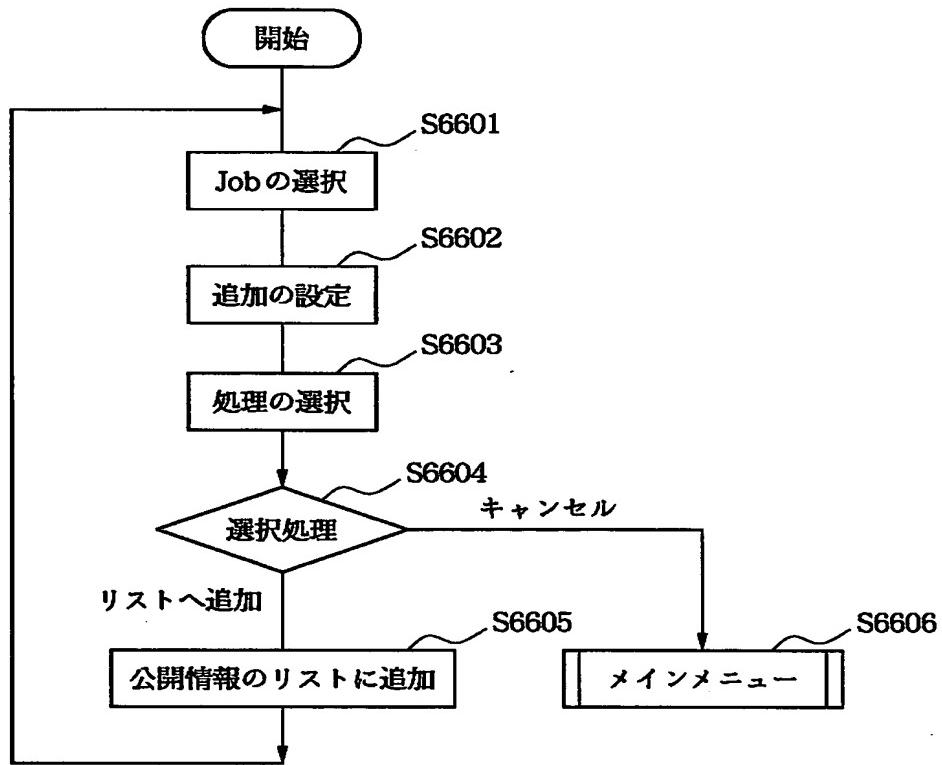
【図64】



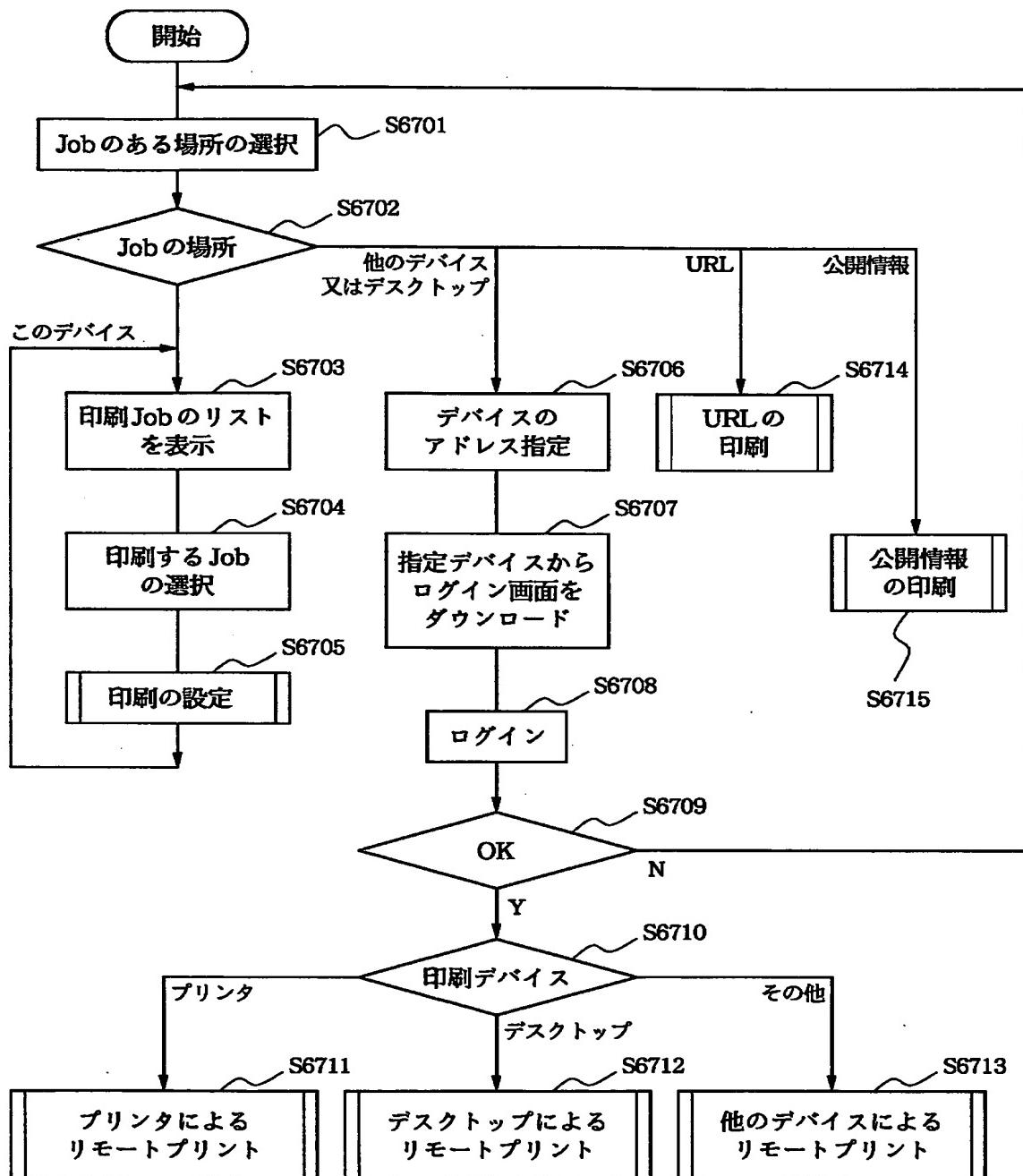
【図65】



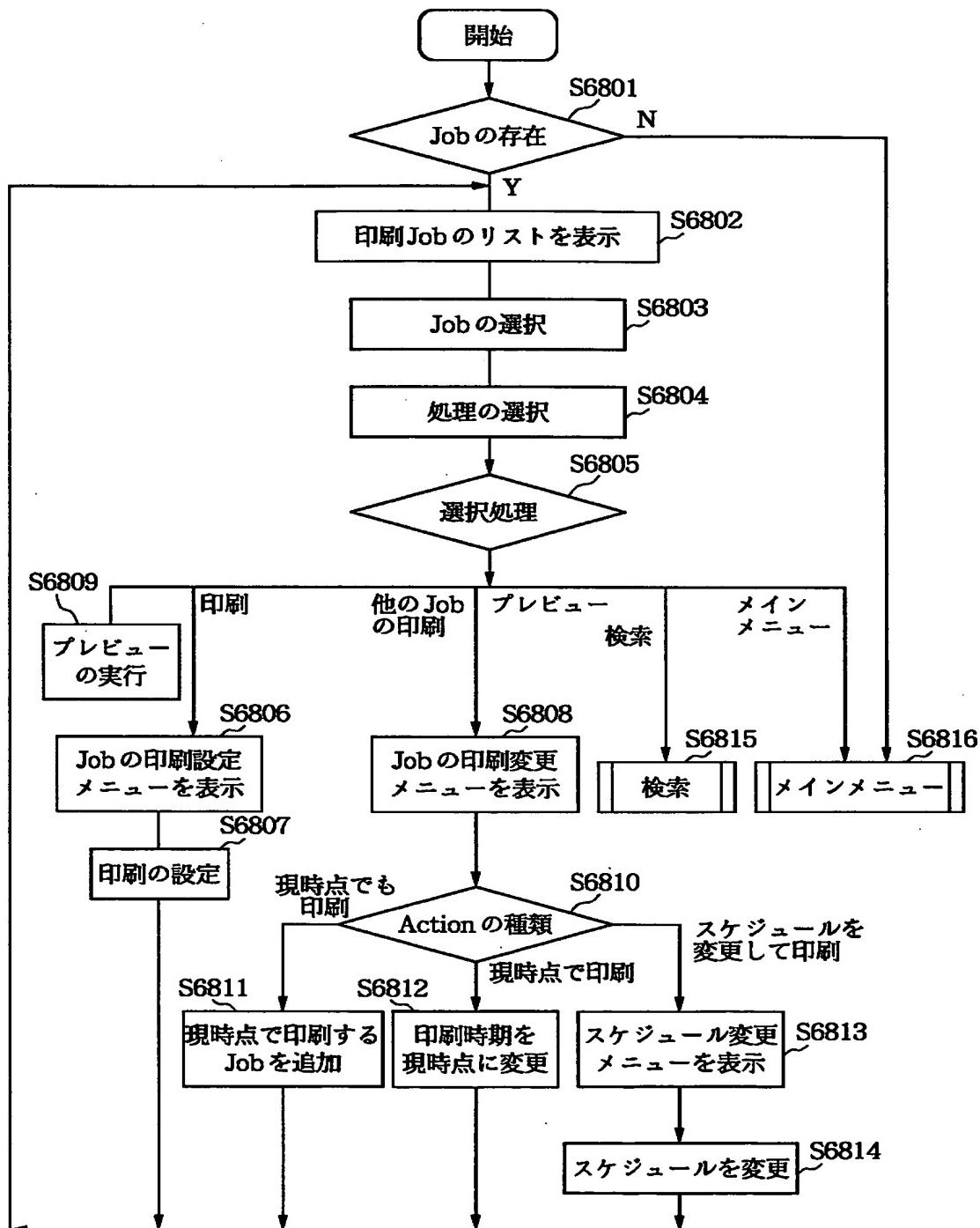
【図66】



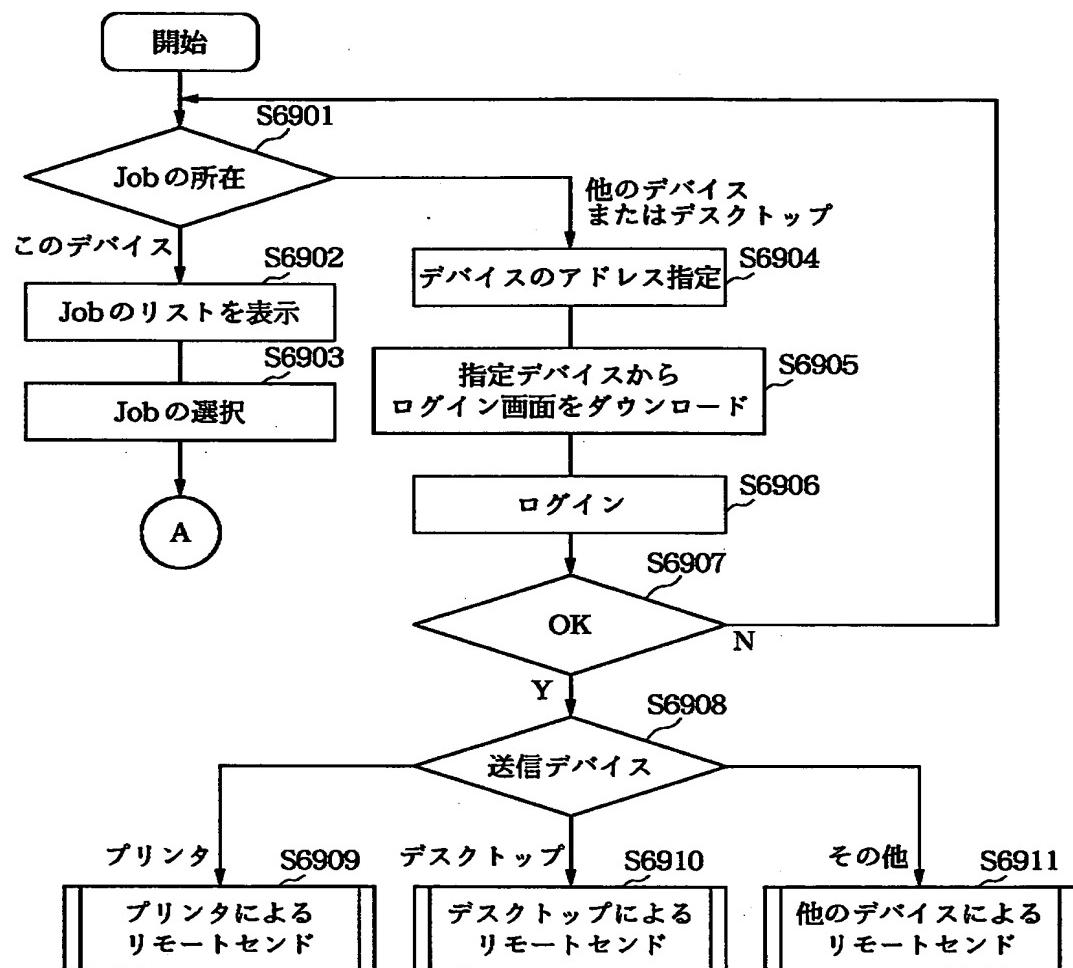
【図67】



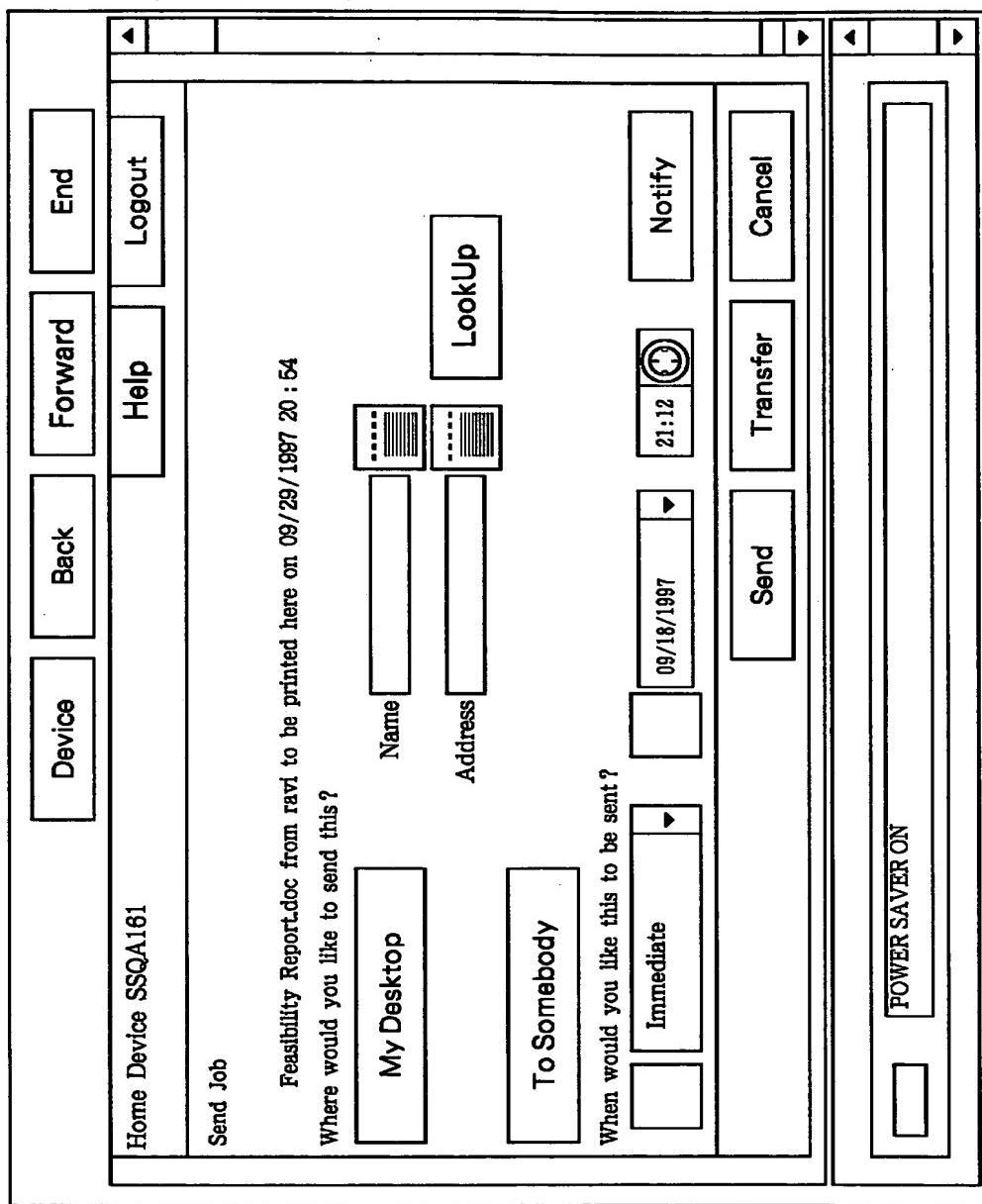
【図68】



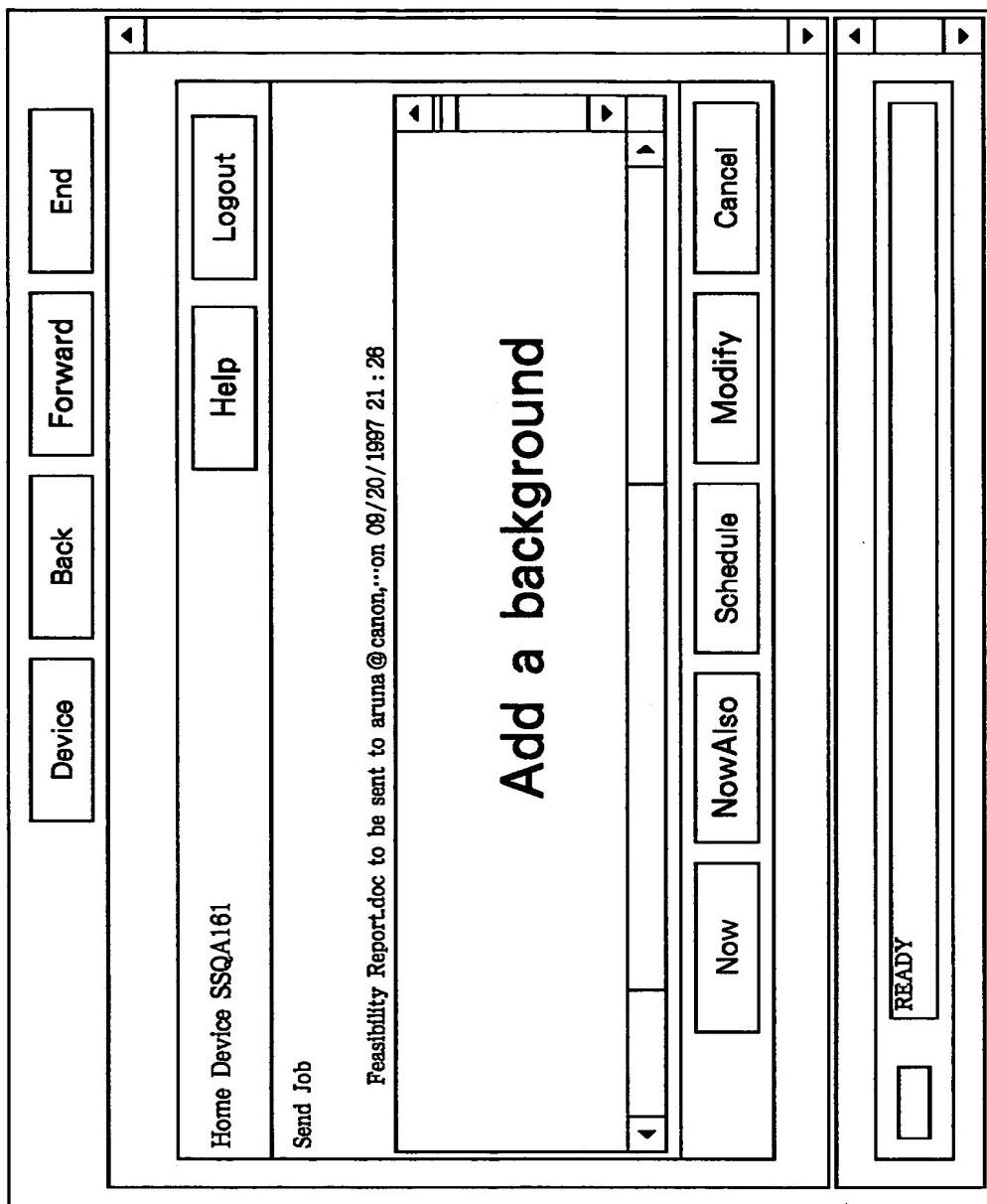
【図69】



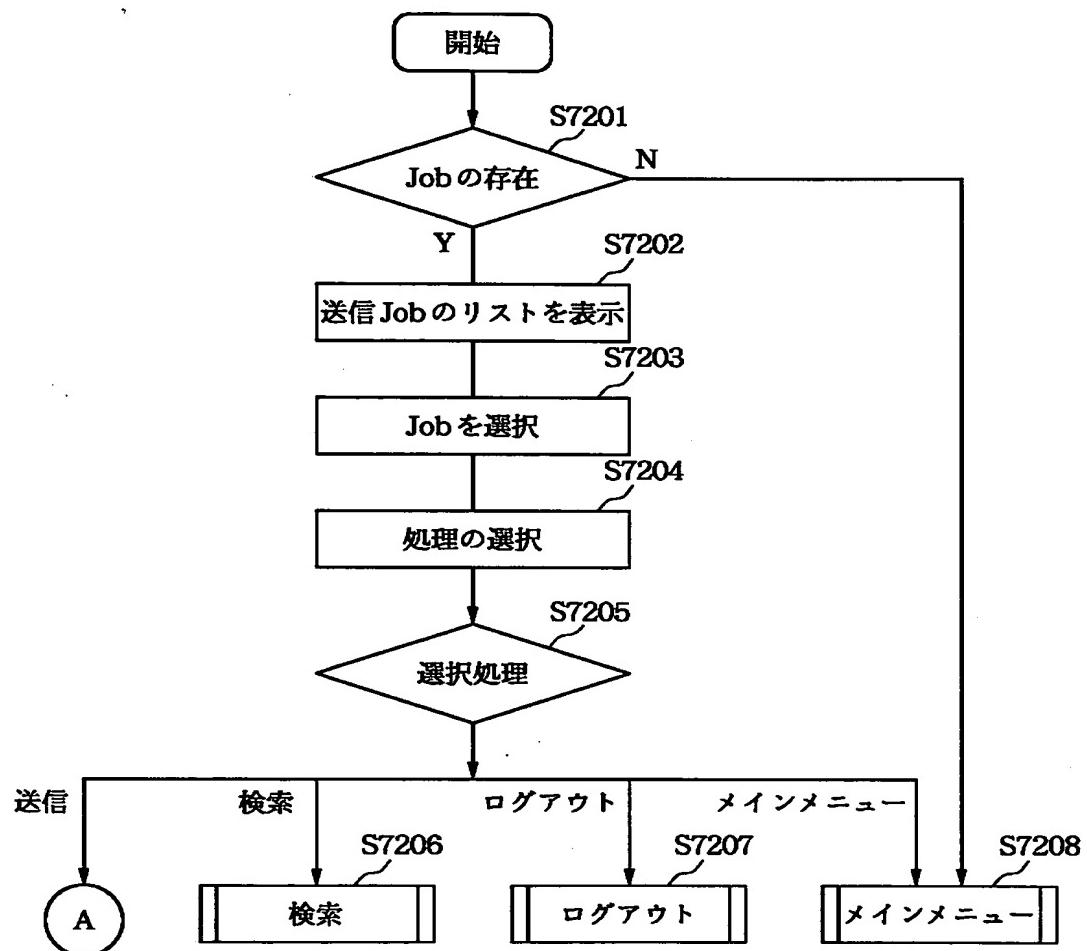
【図70】



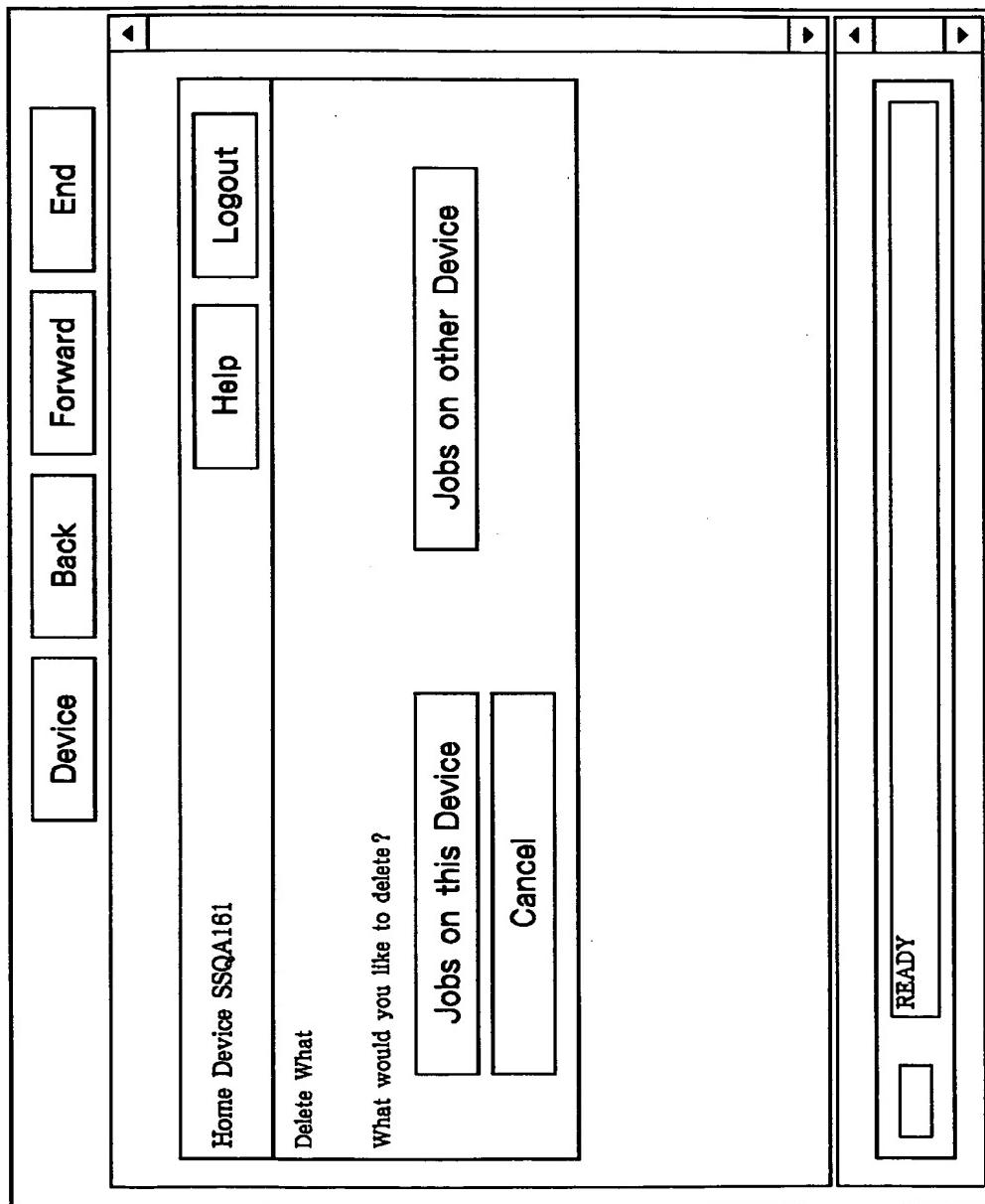
【図71】



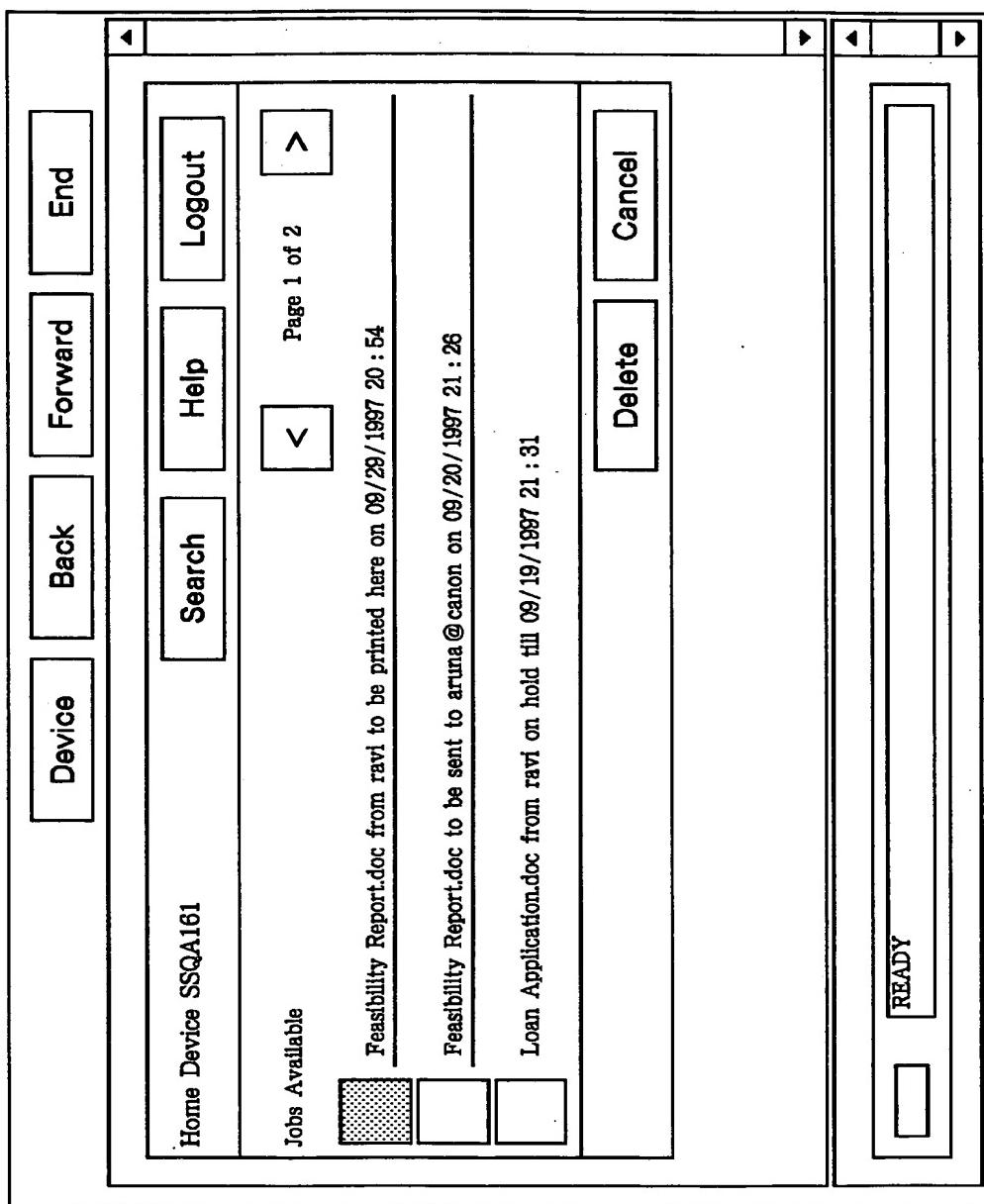
【図72】



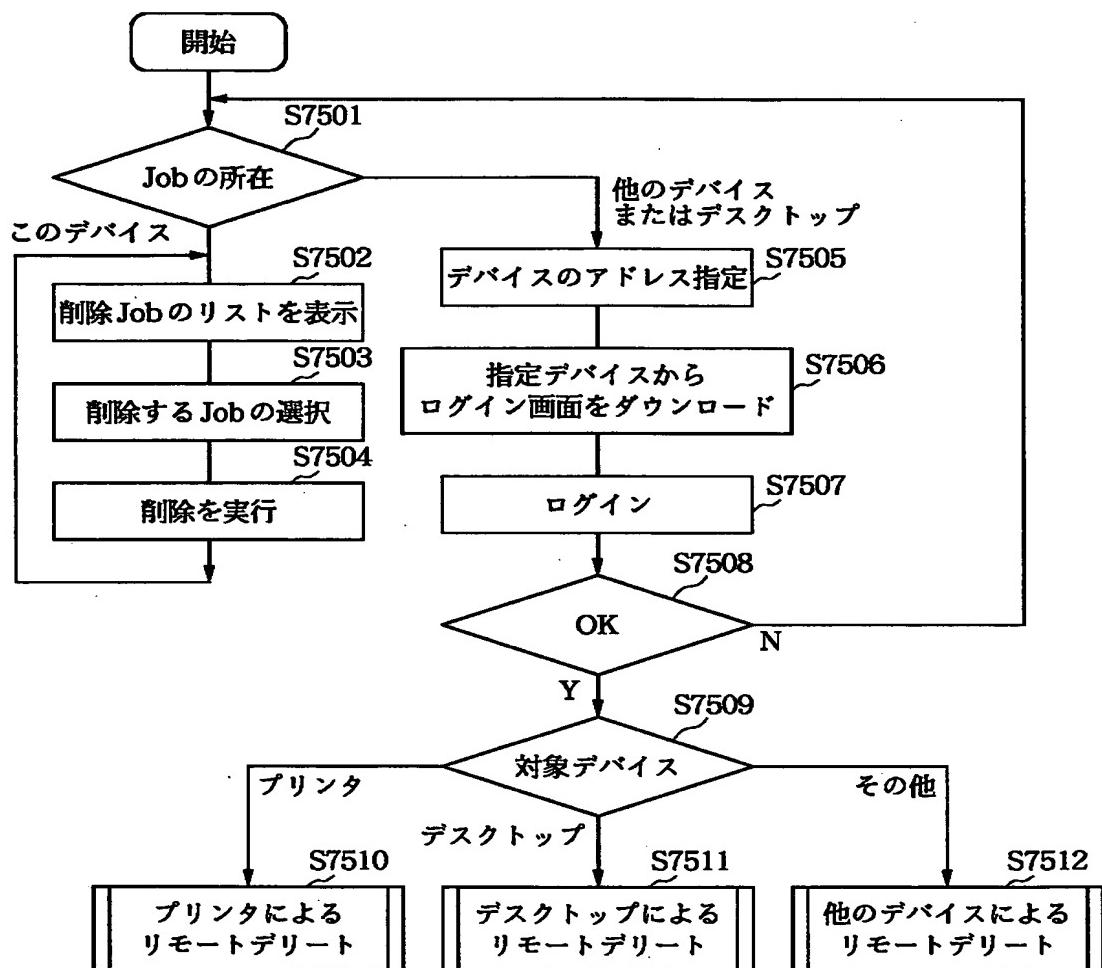
【図73】



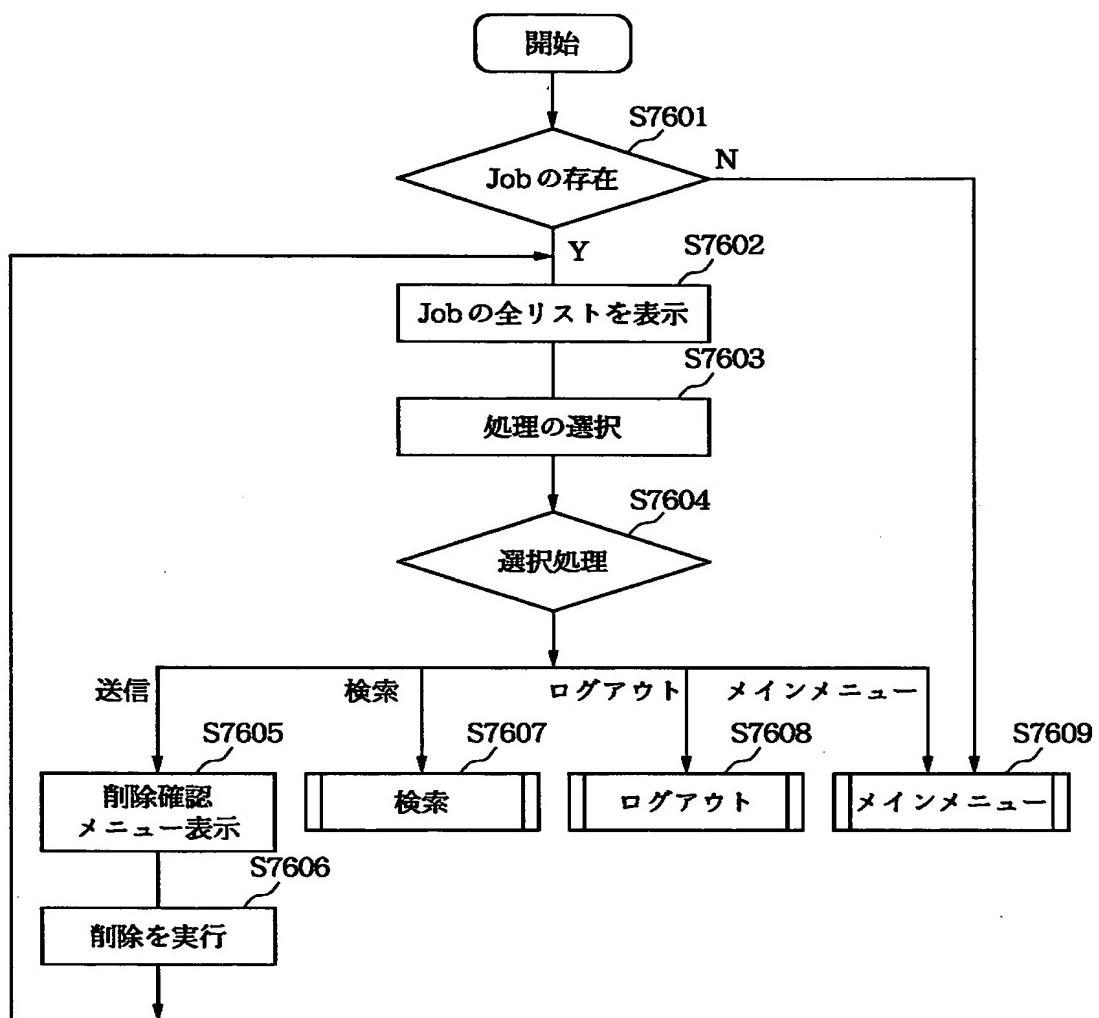
【図74】



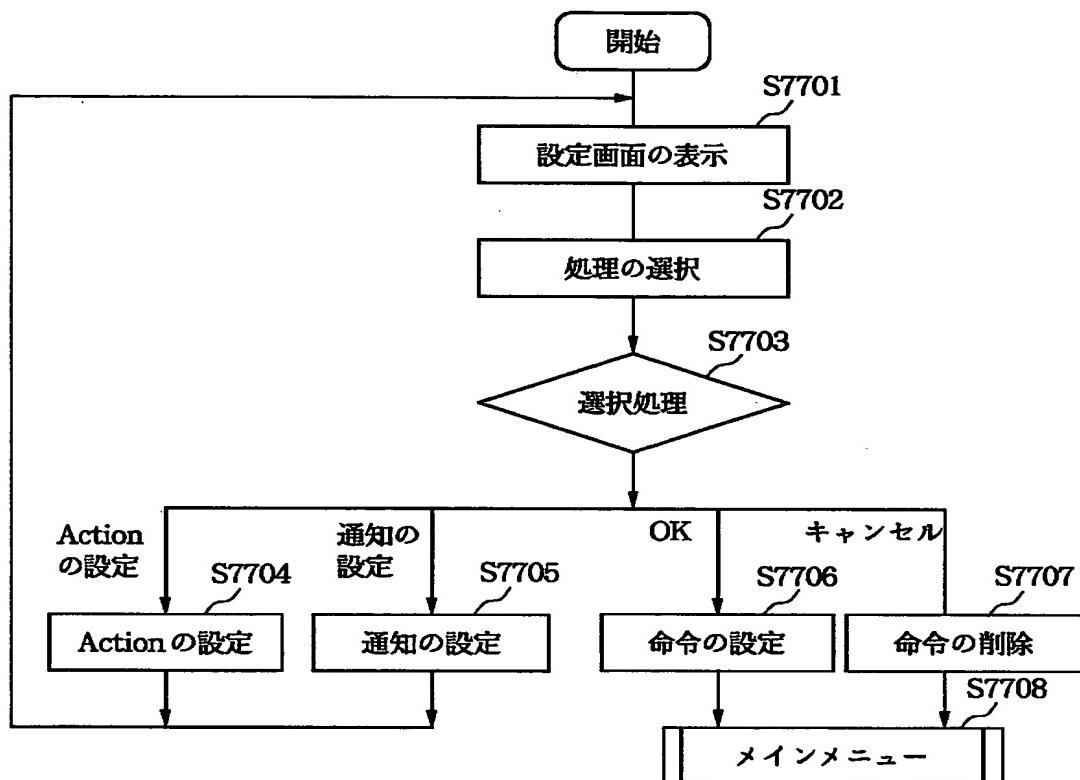
【図75】



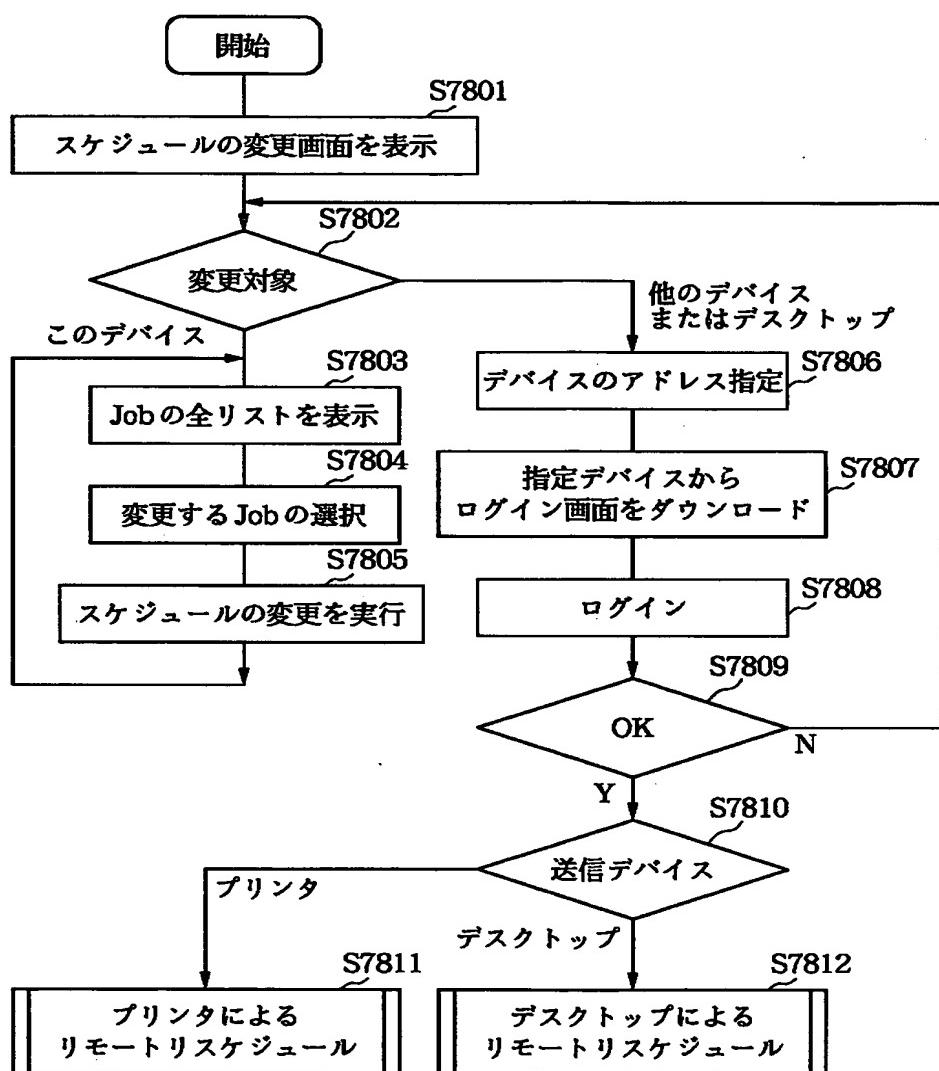
【図76】



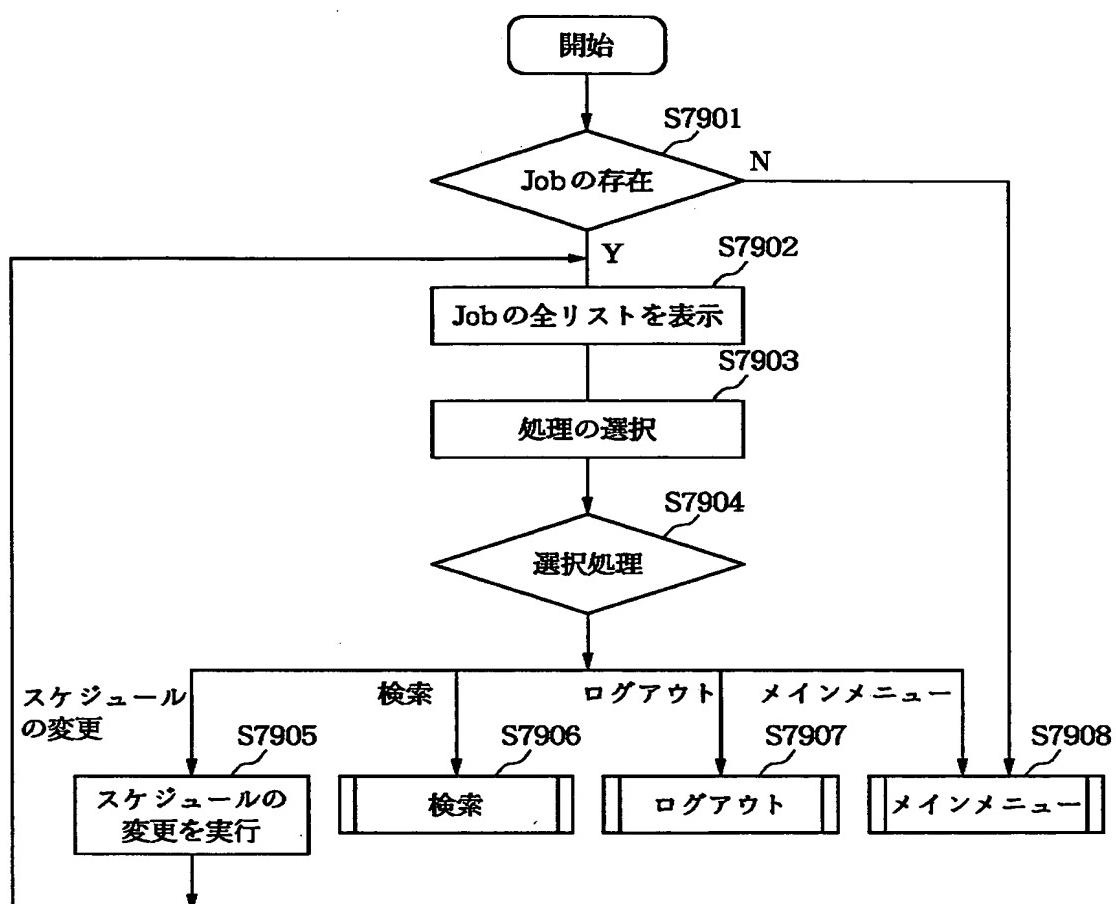
【図77】



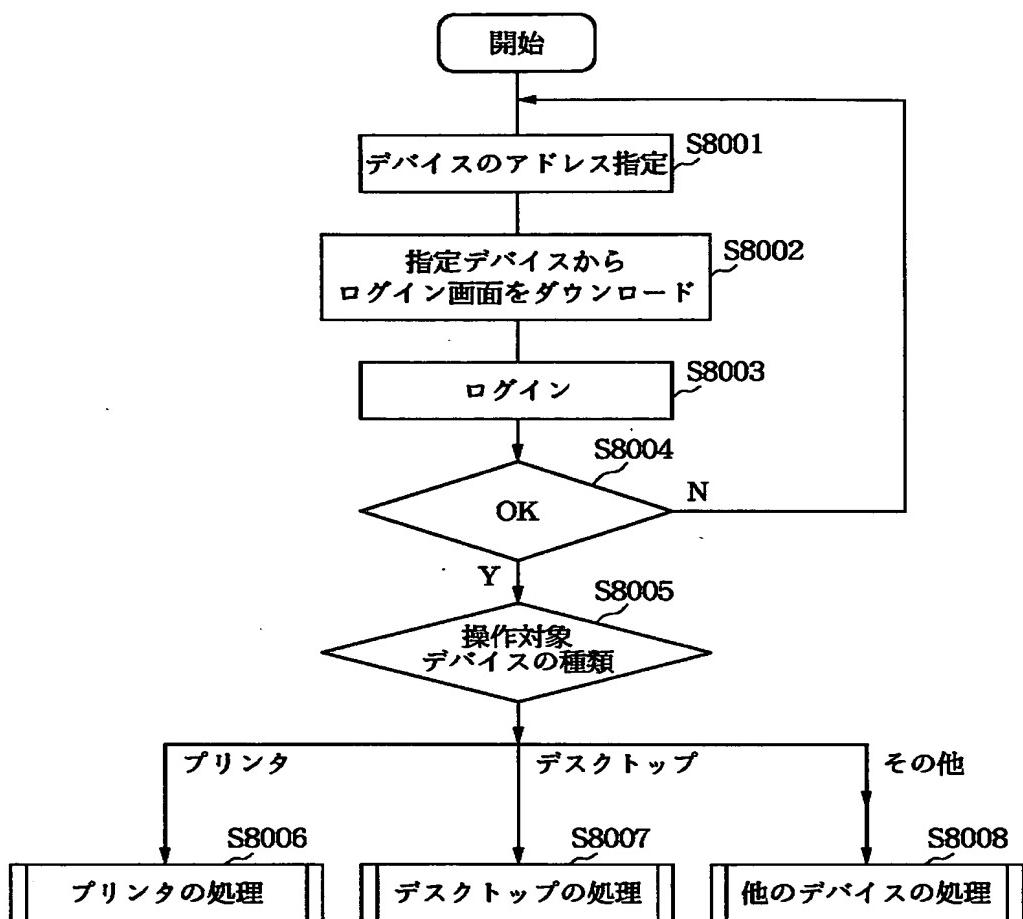
【図78】



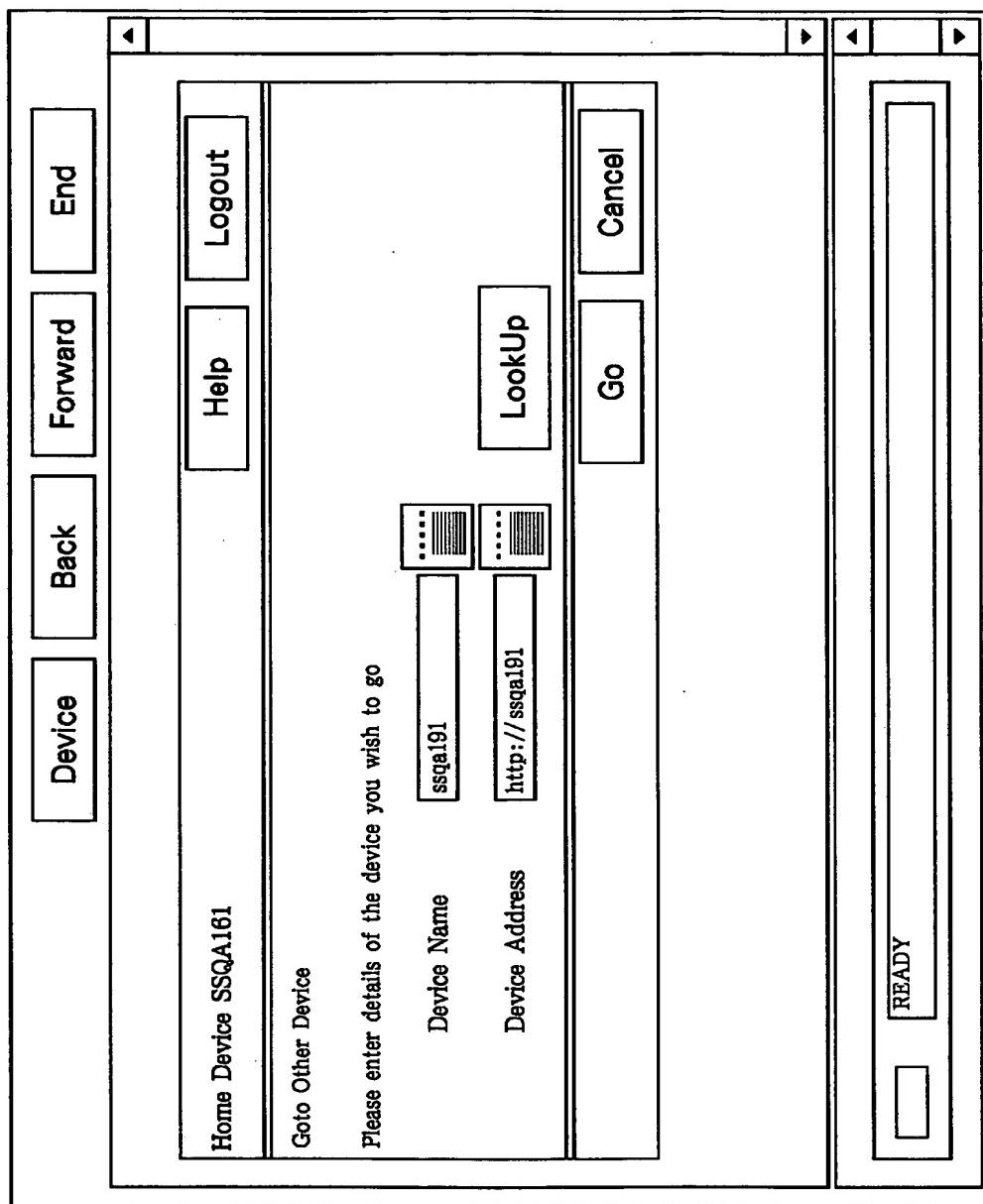
【図79】



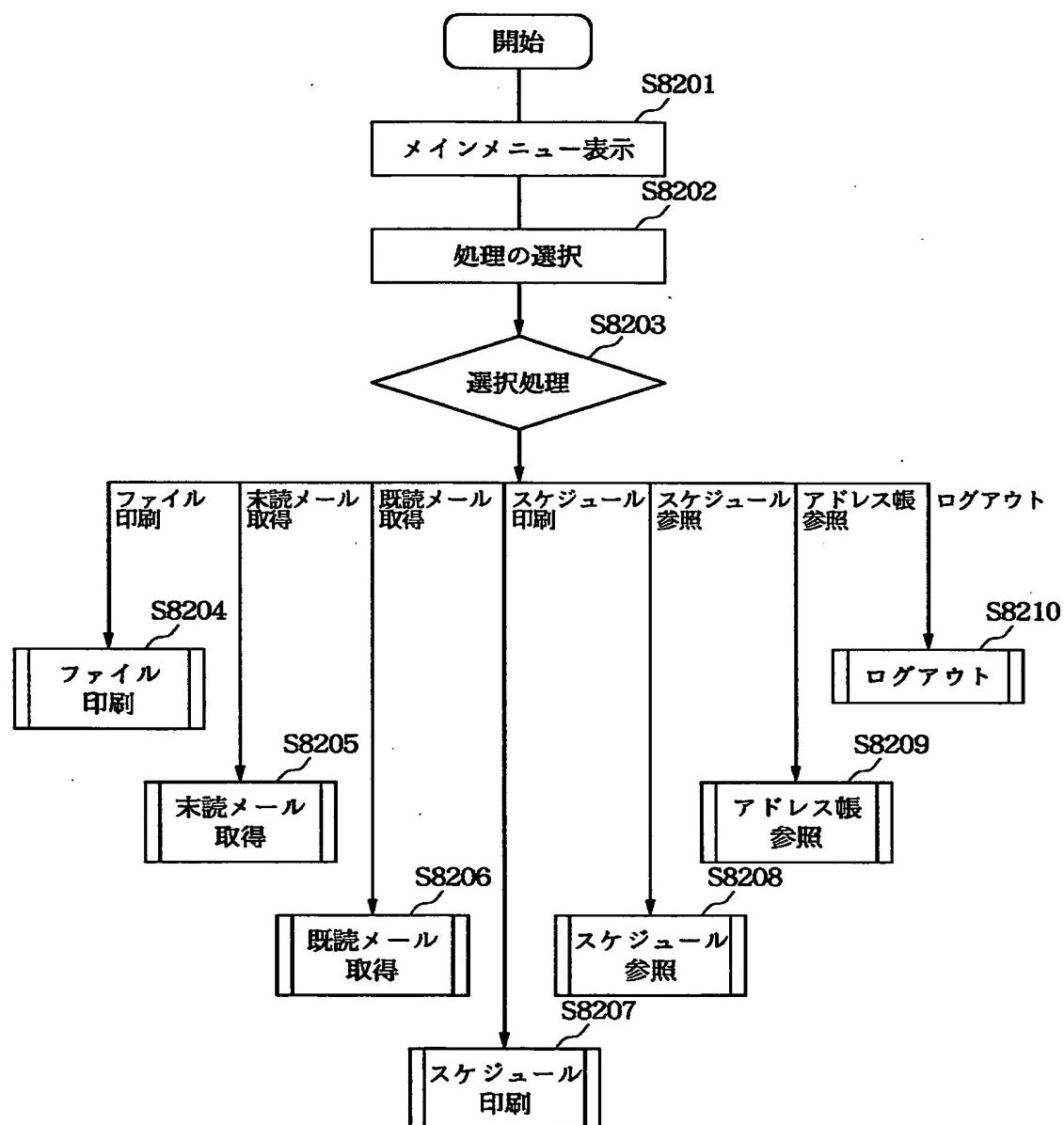
【図80】



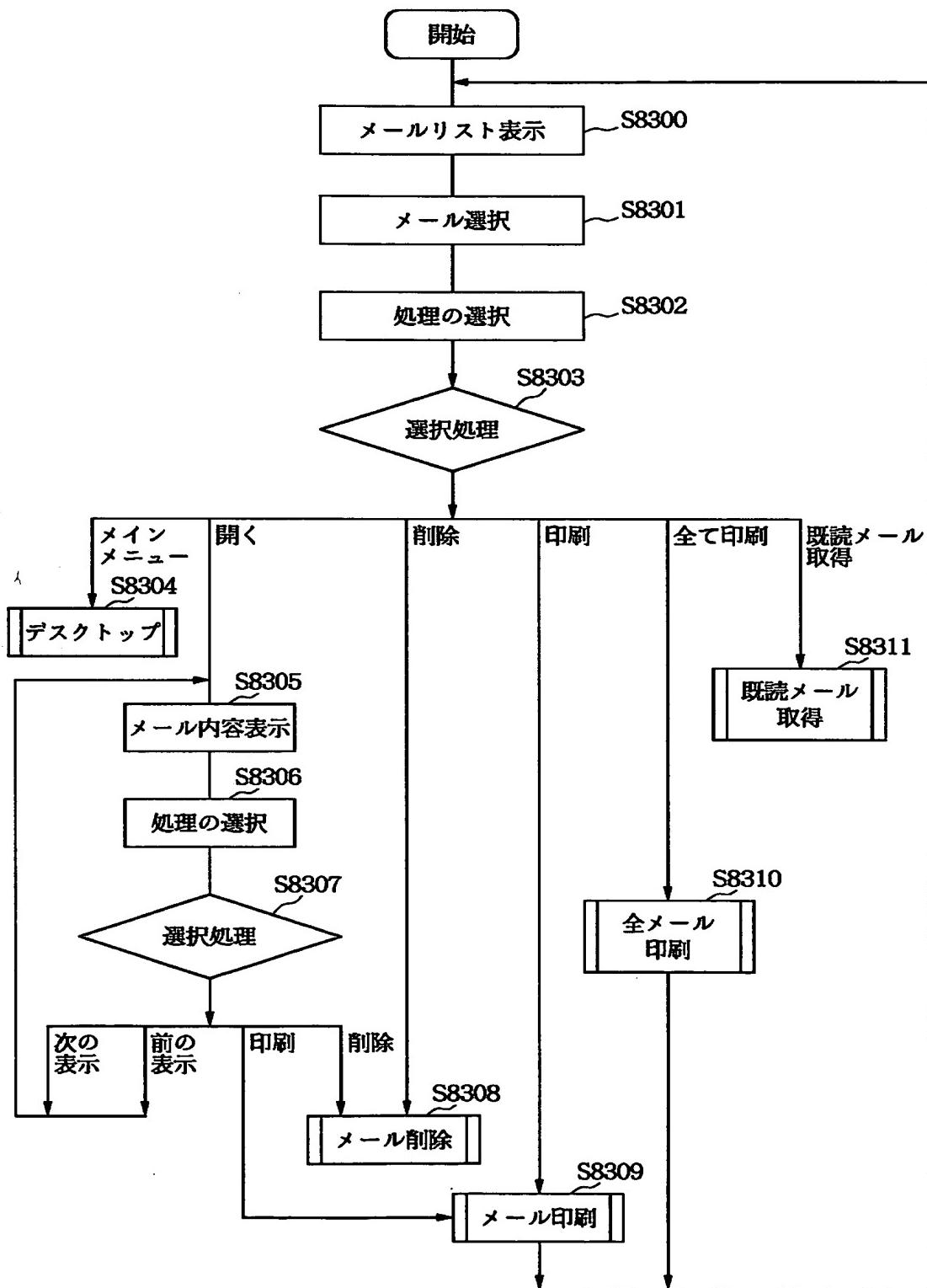
【図81】



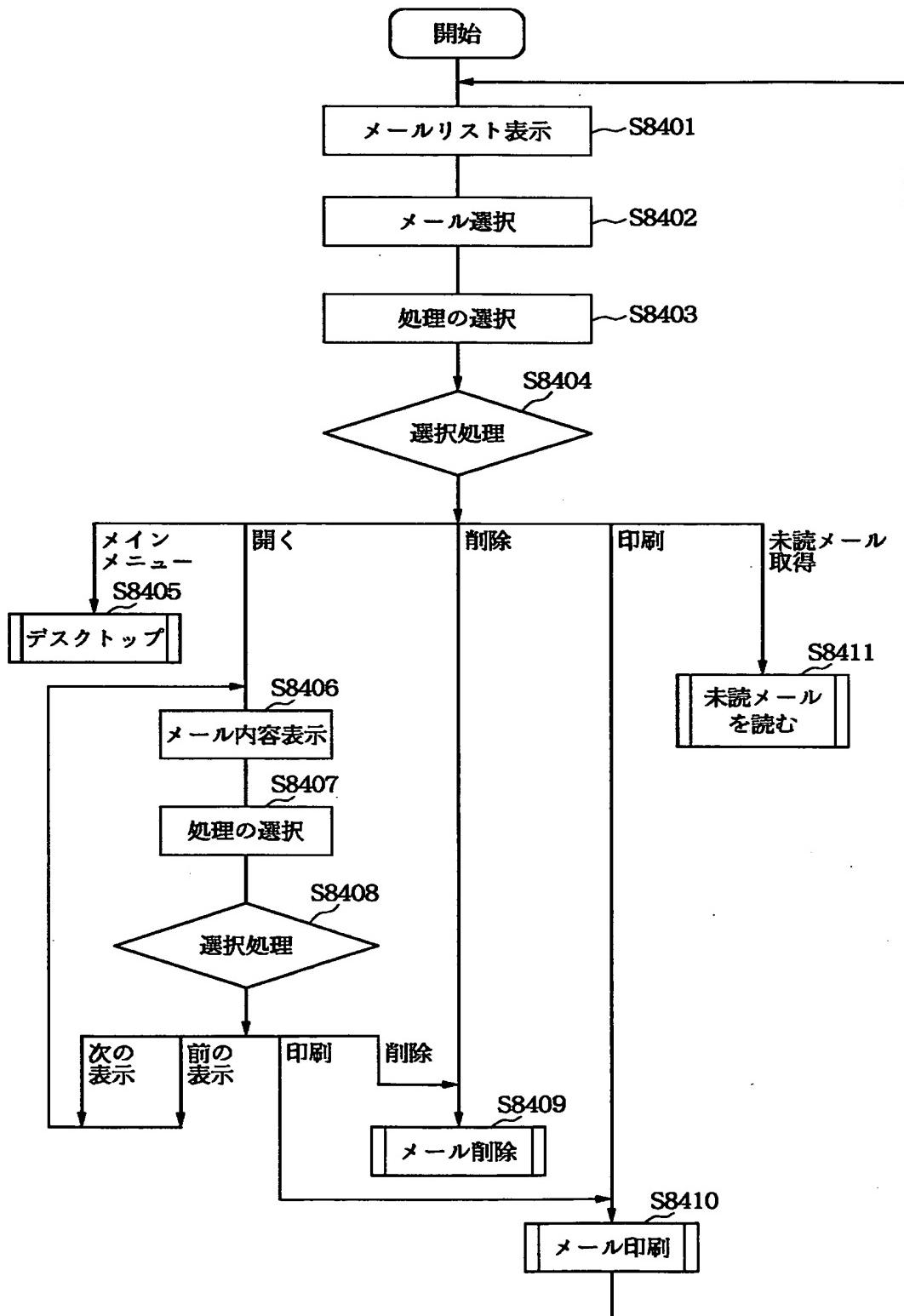
【図82】



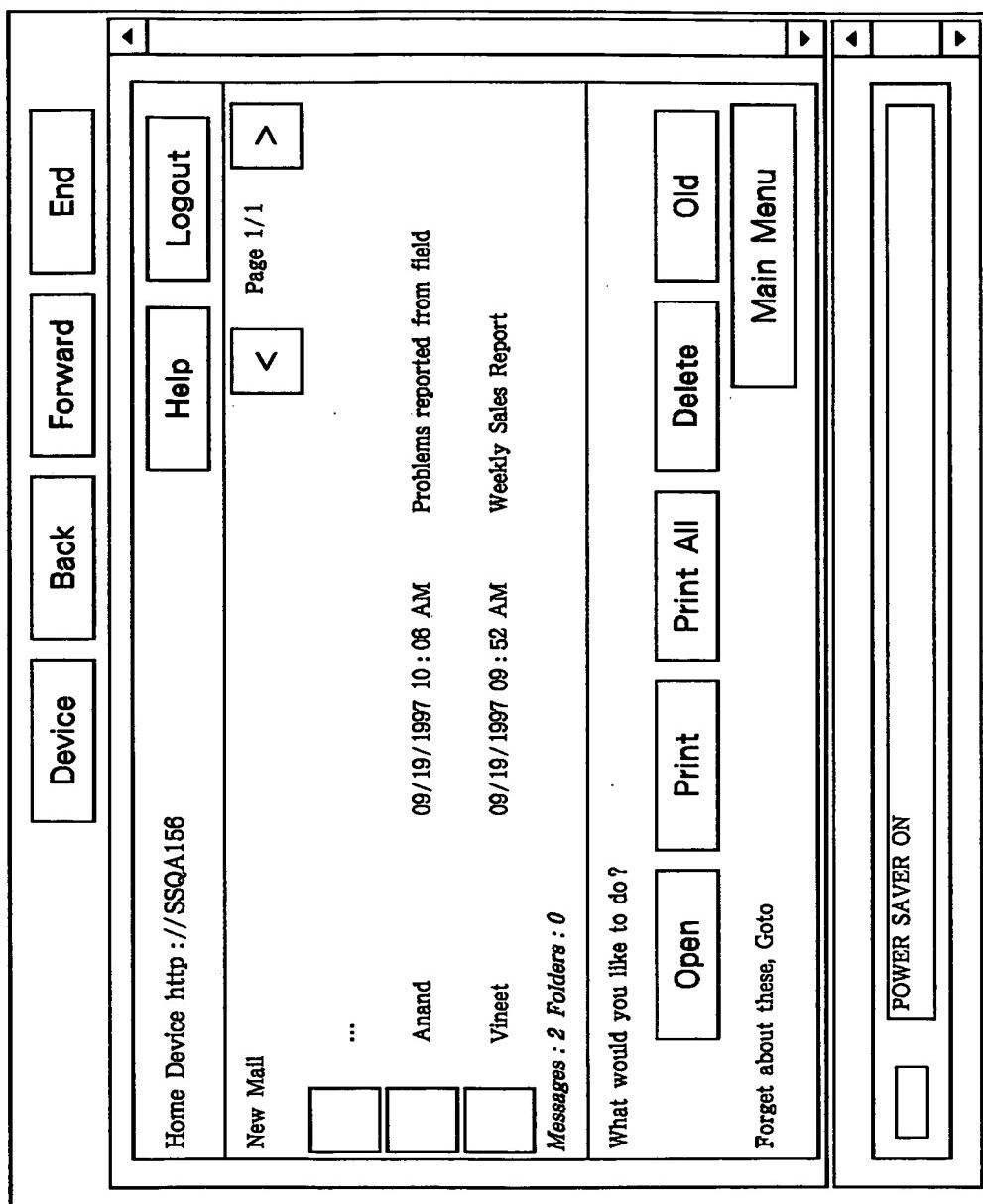
【図83】



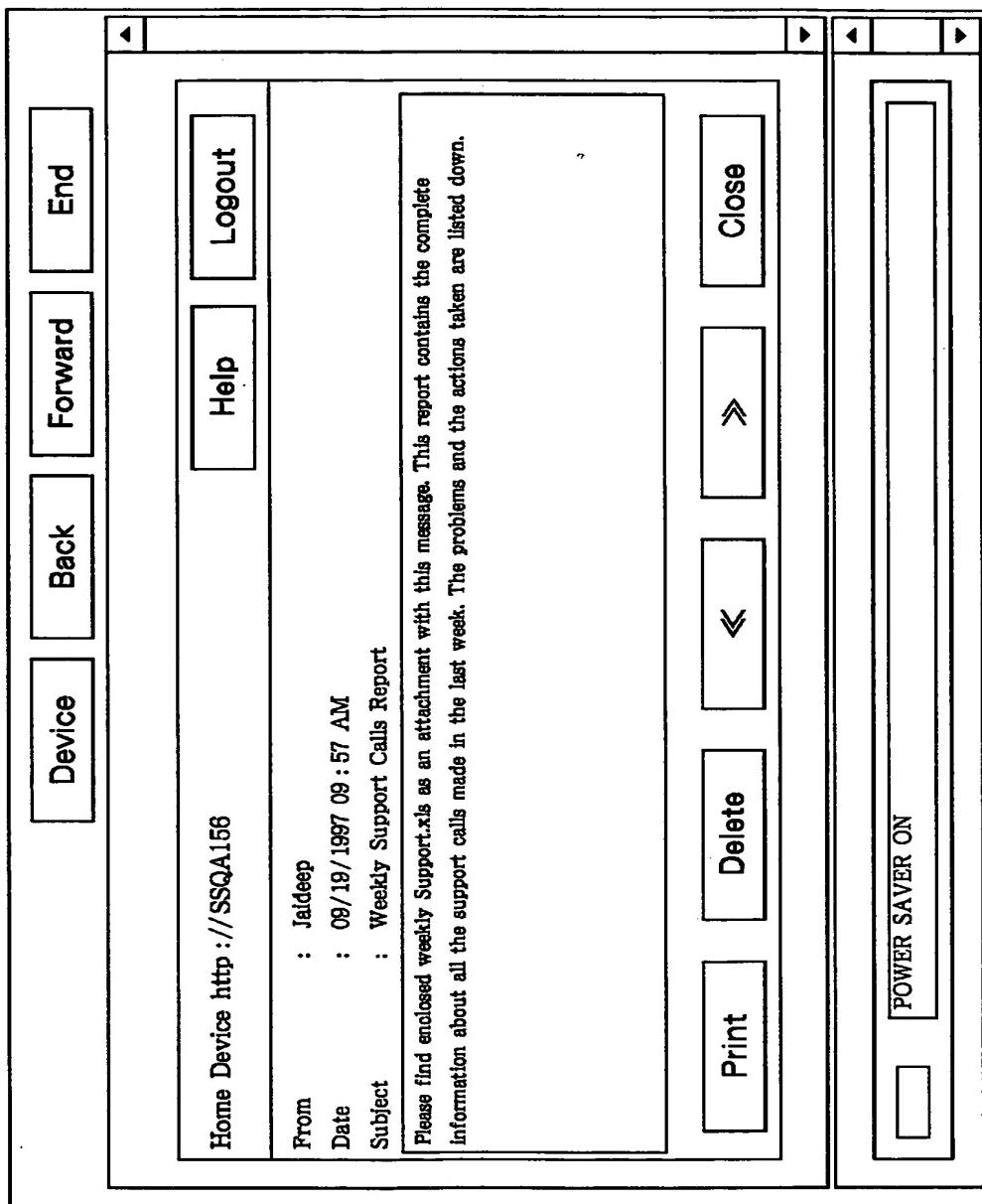
【図84】



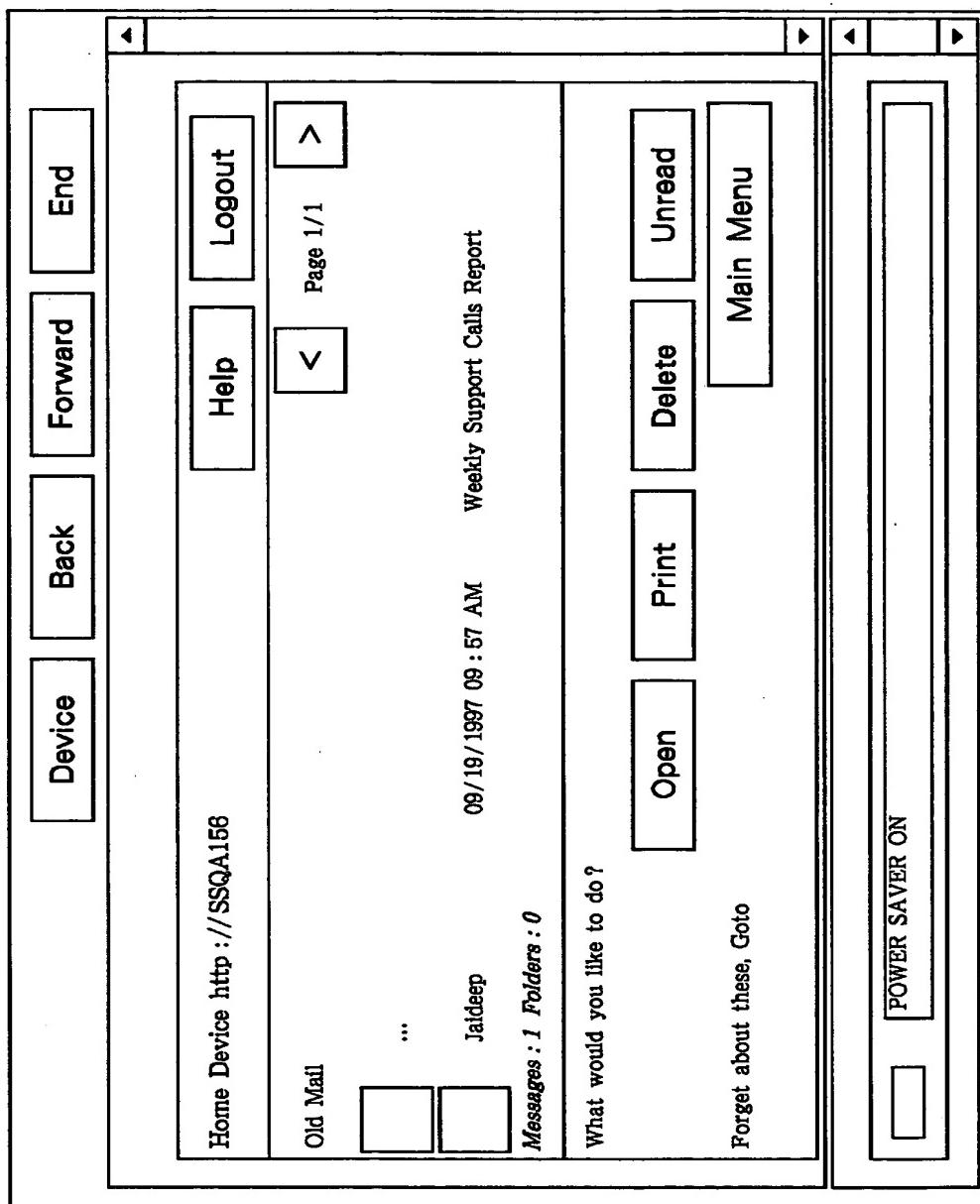
【図85】



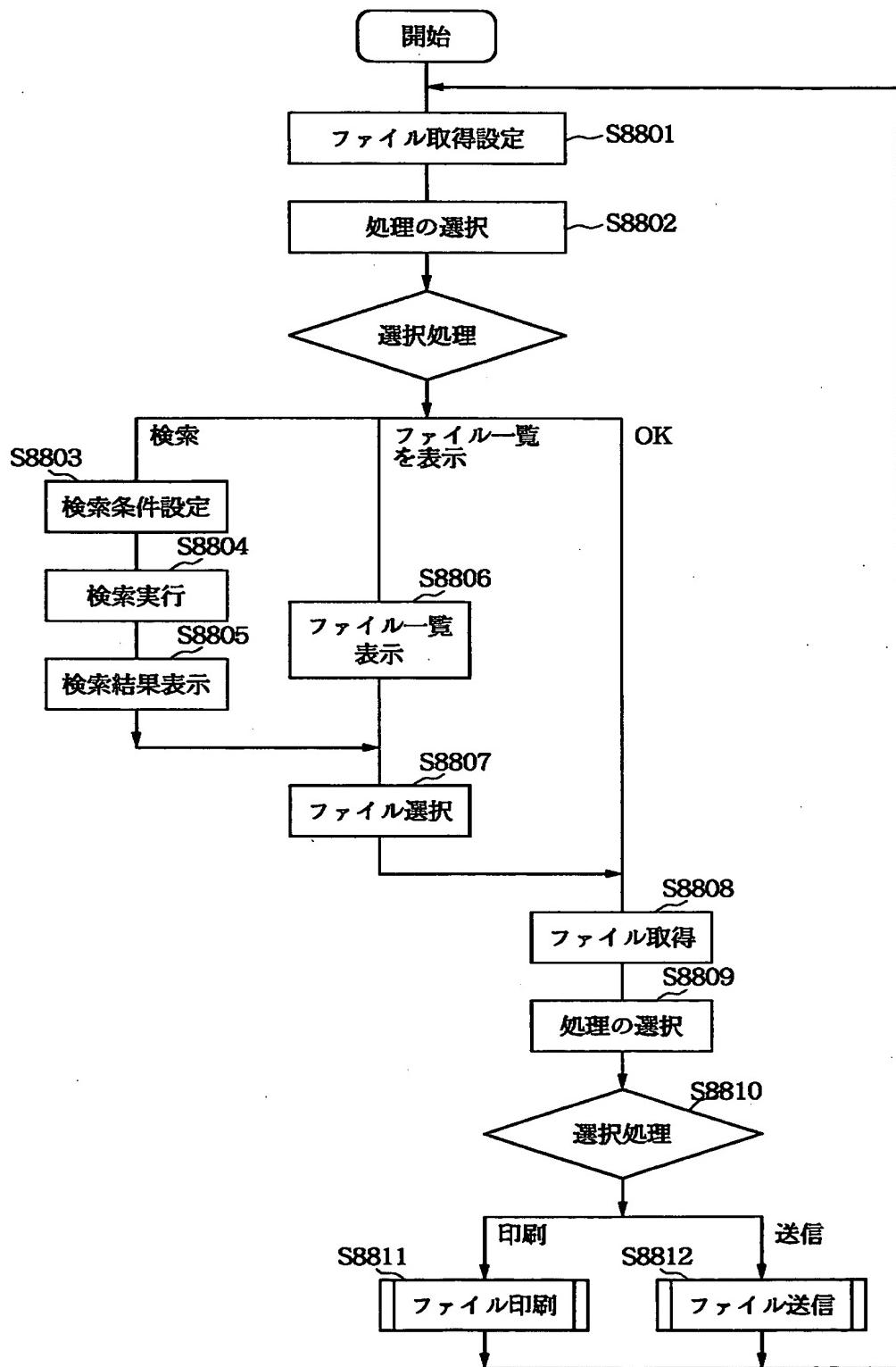
【図86】



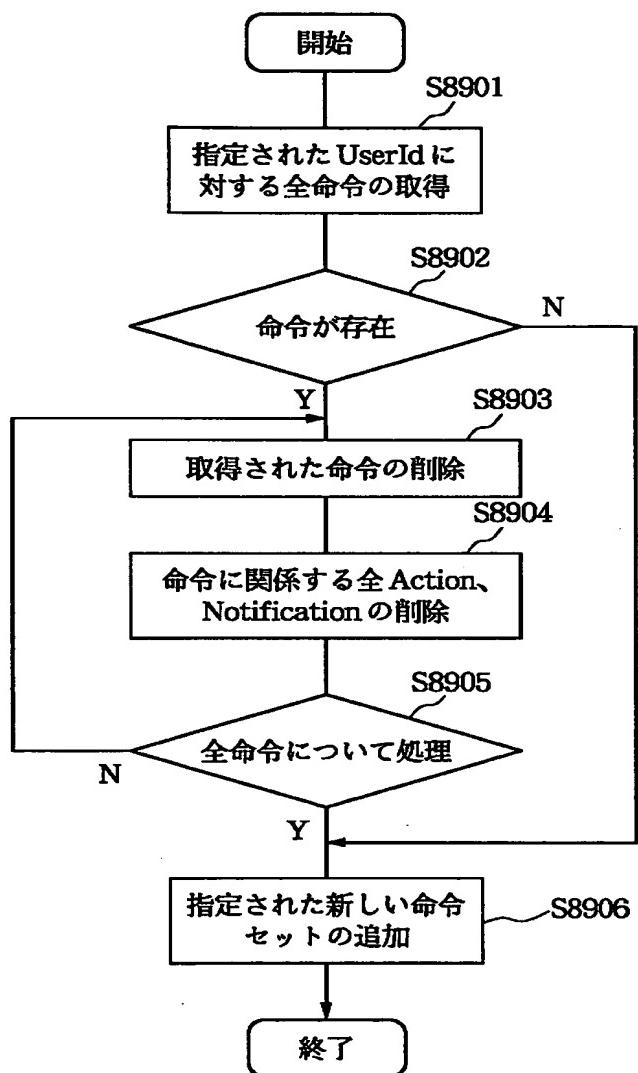
【図 87】



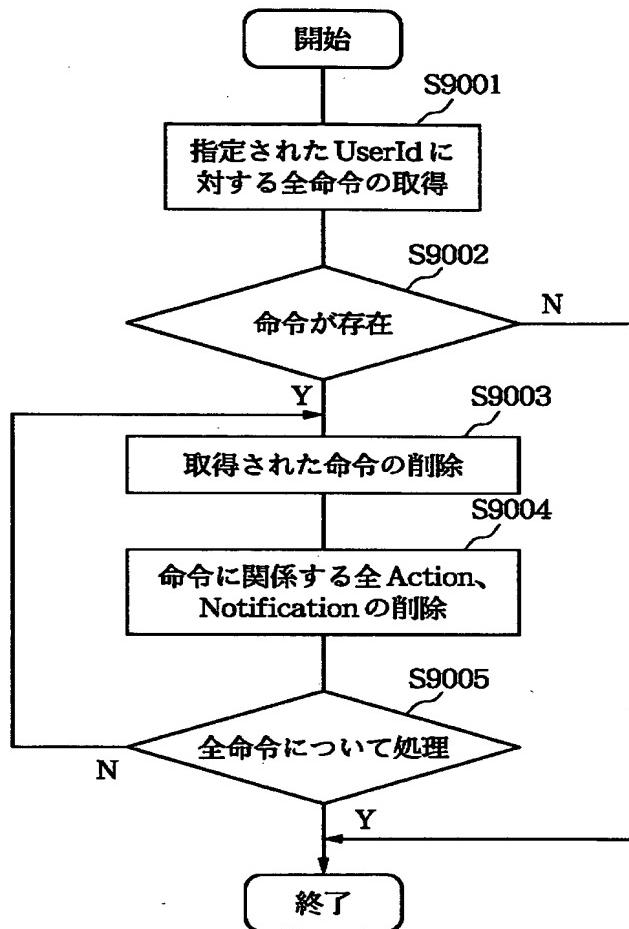
【図 8.8】



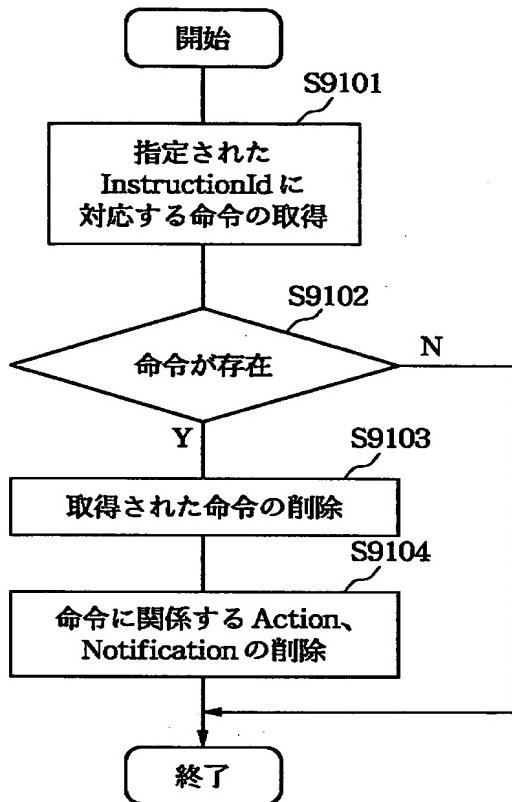
【図89】



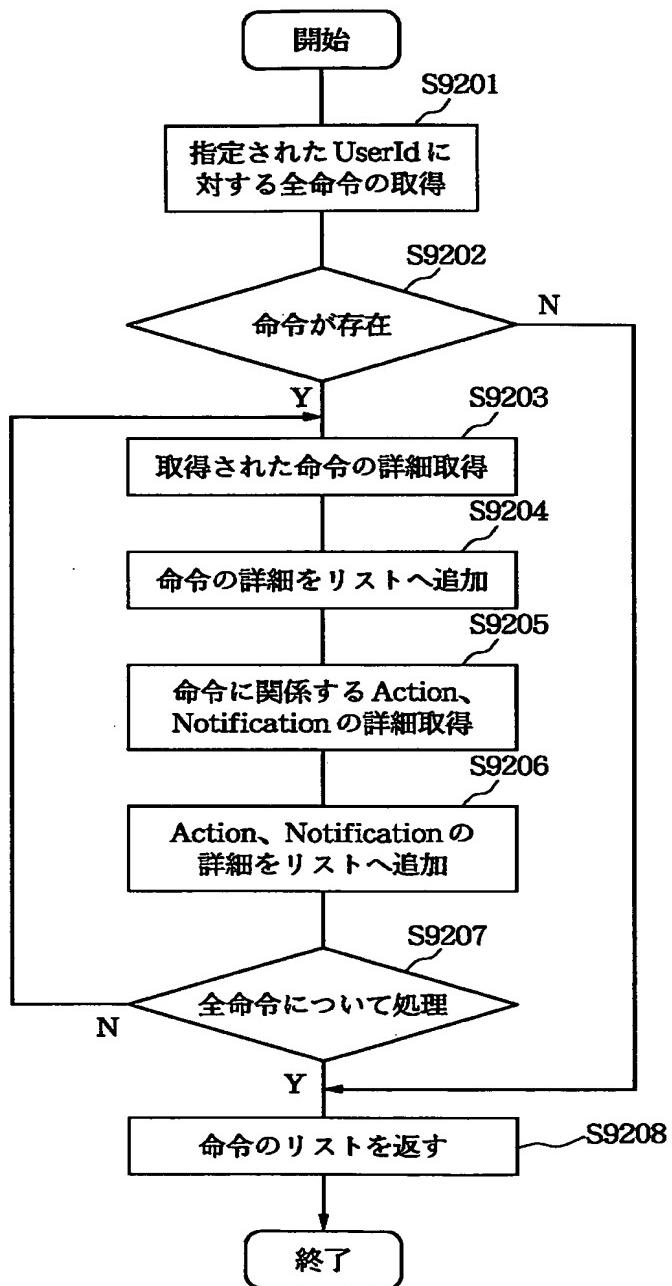
【図90】



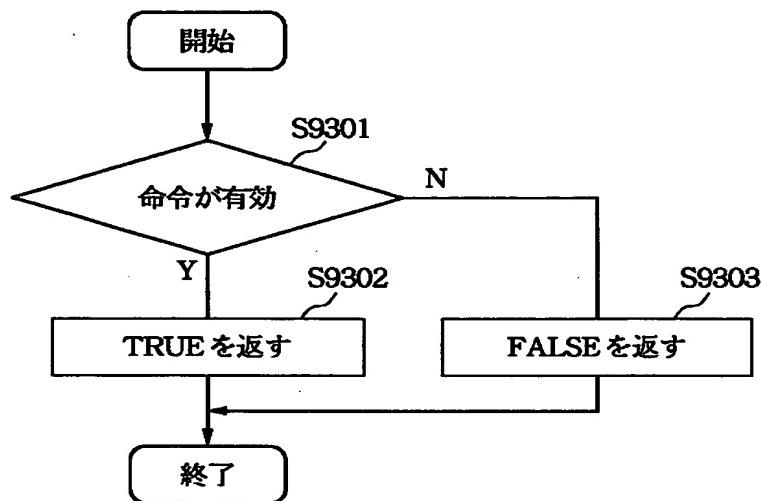
【図91】



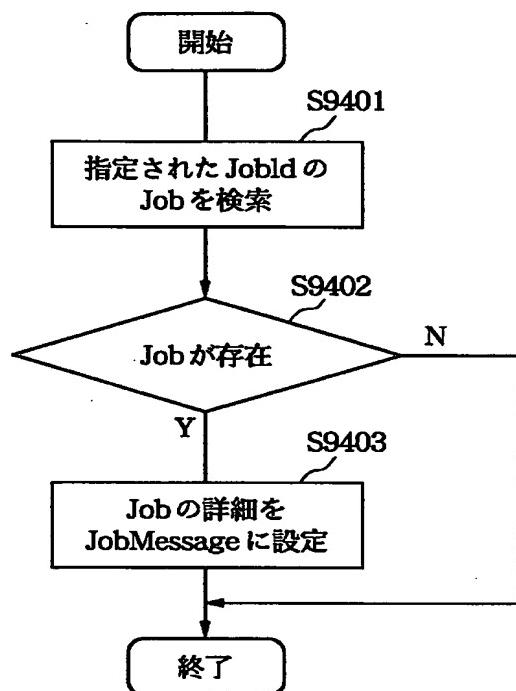
【図92】



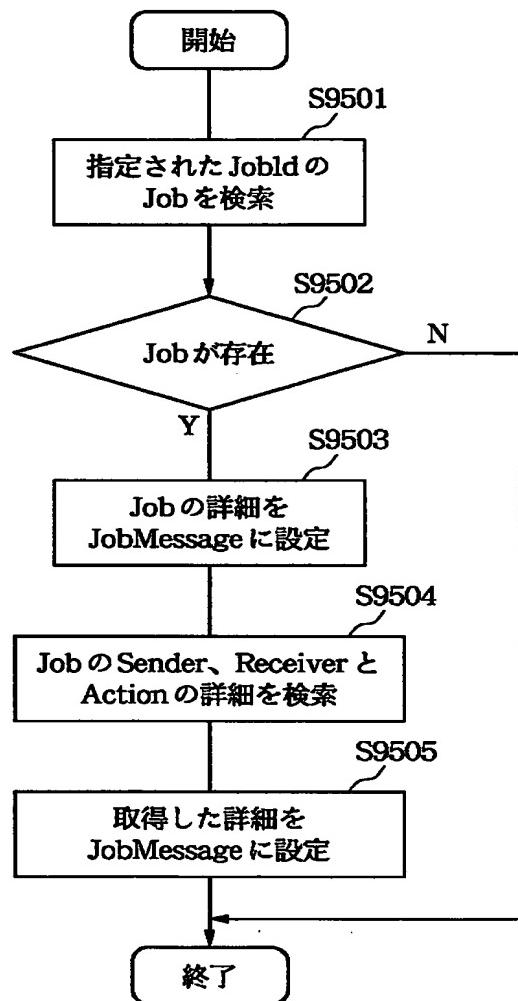
【図93】



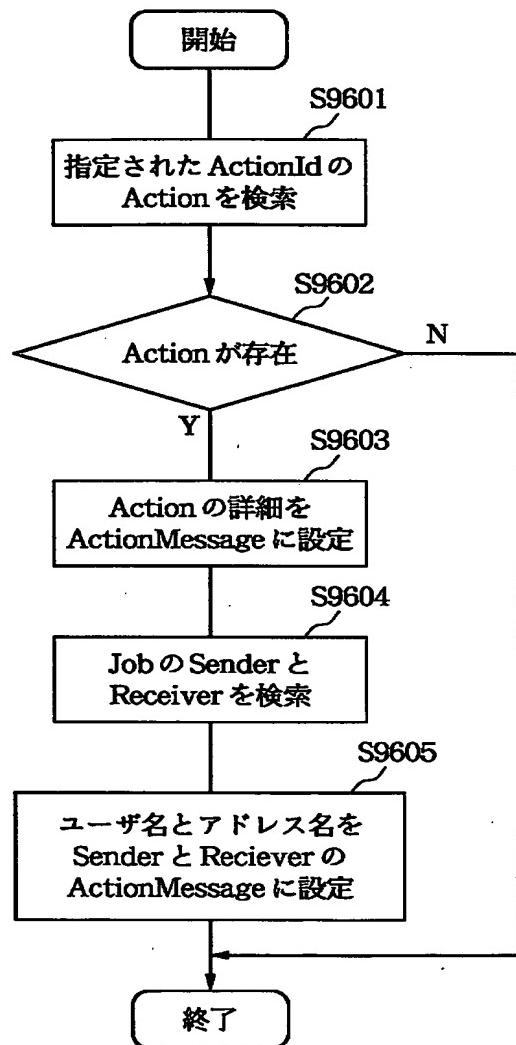
【図94】



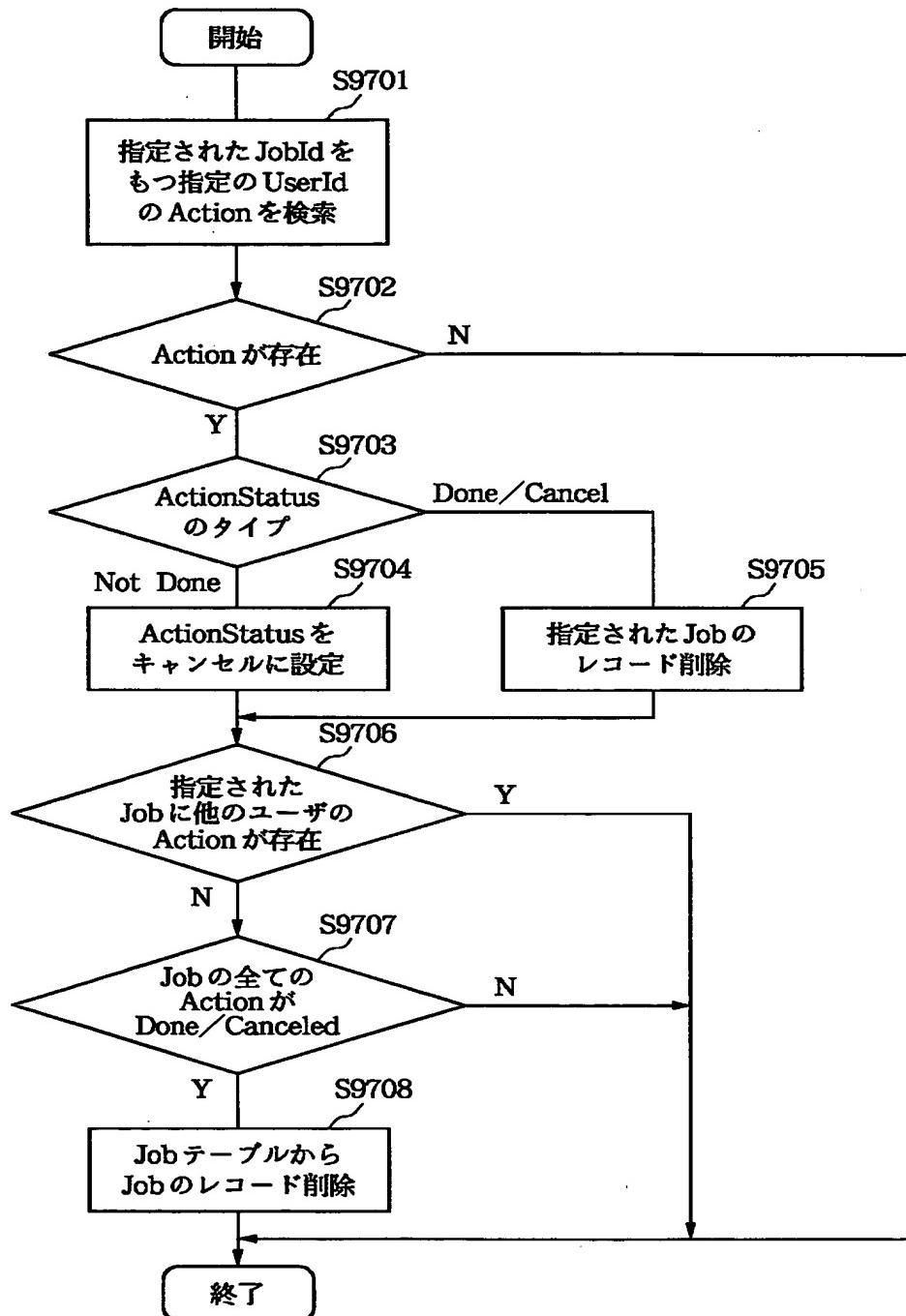
【図95】



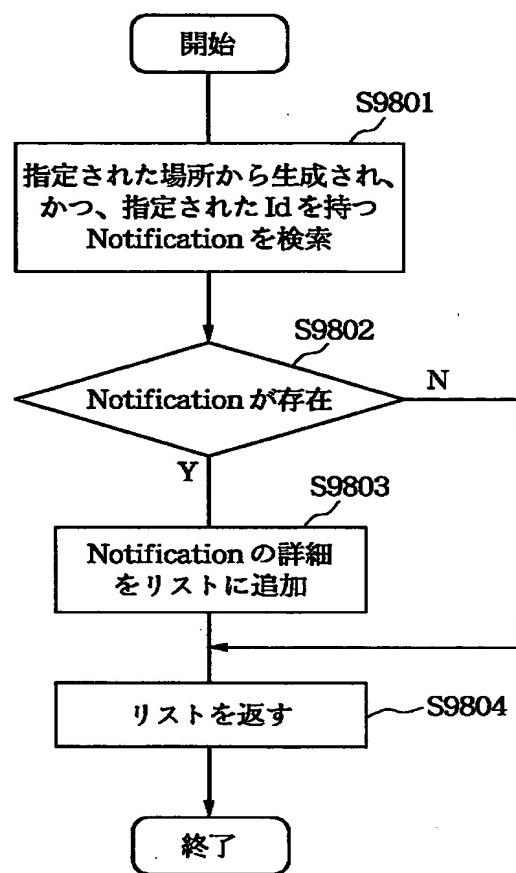
【図96】



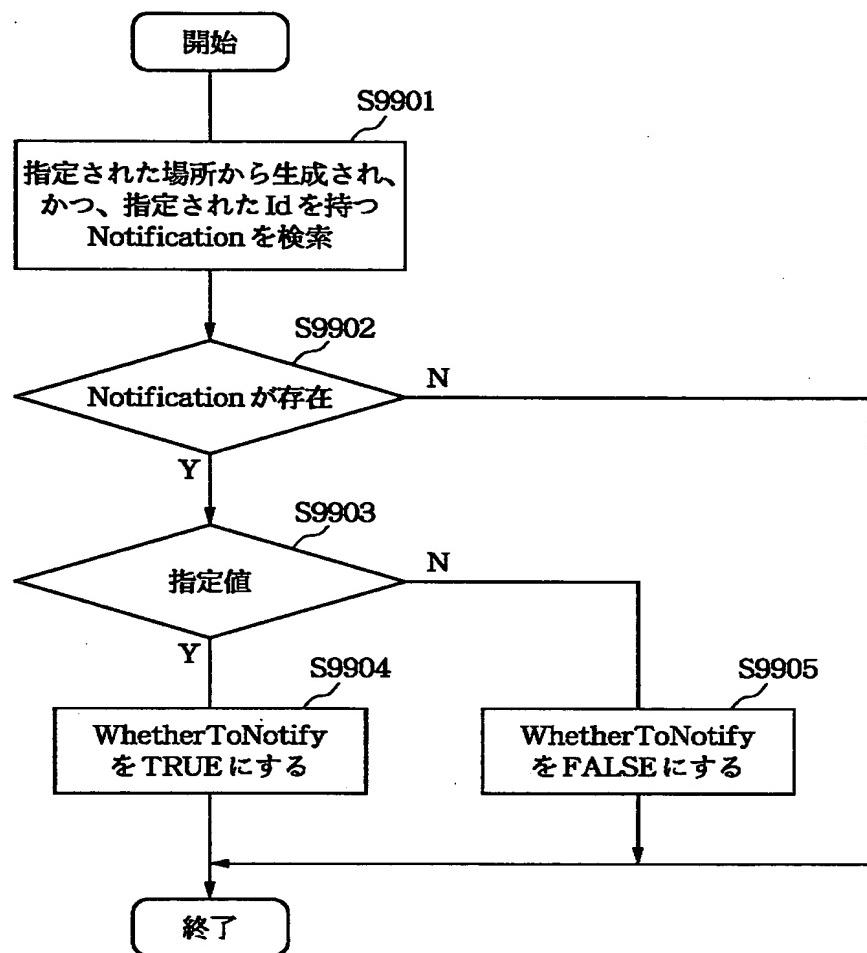
【図97】



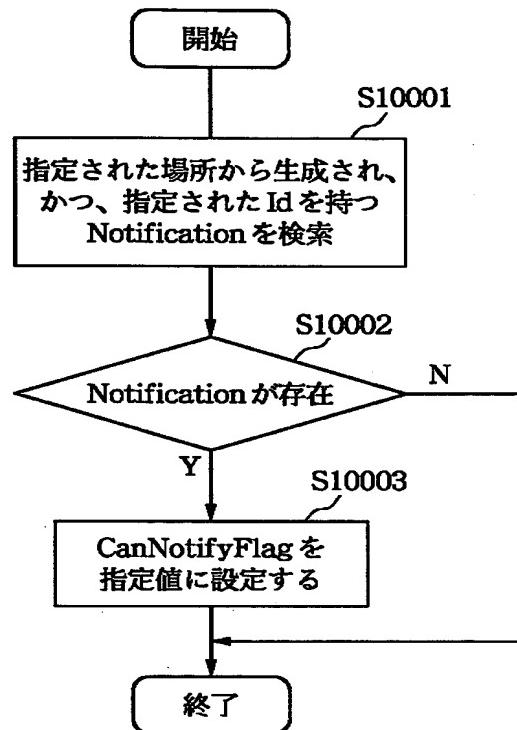
【図98】



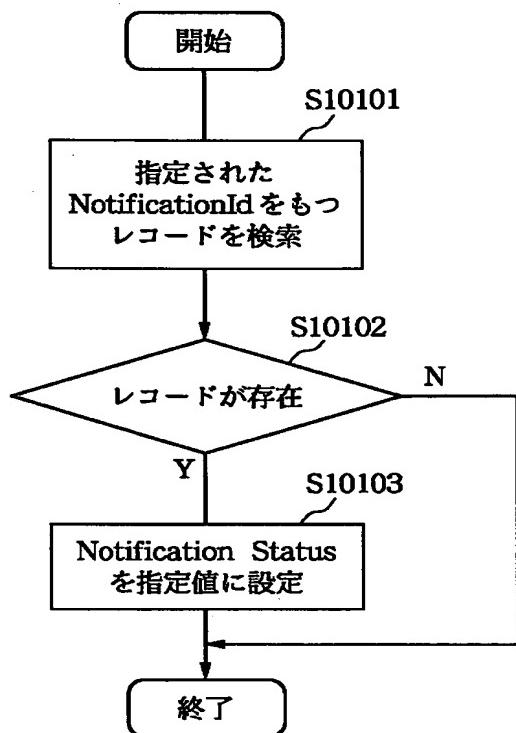
【図99】



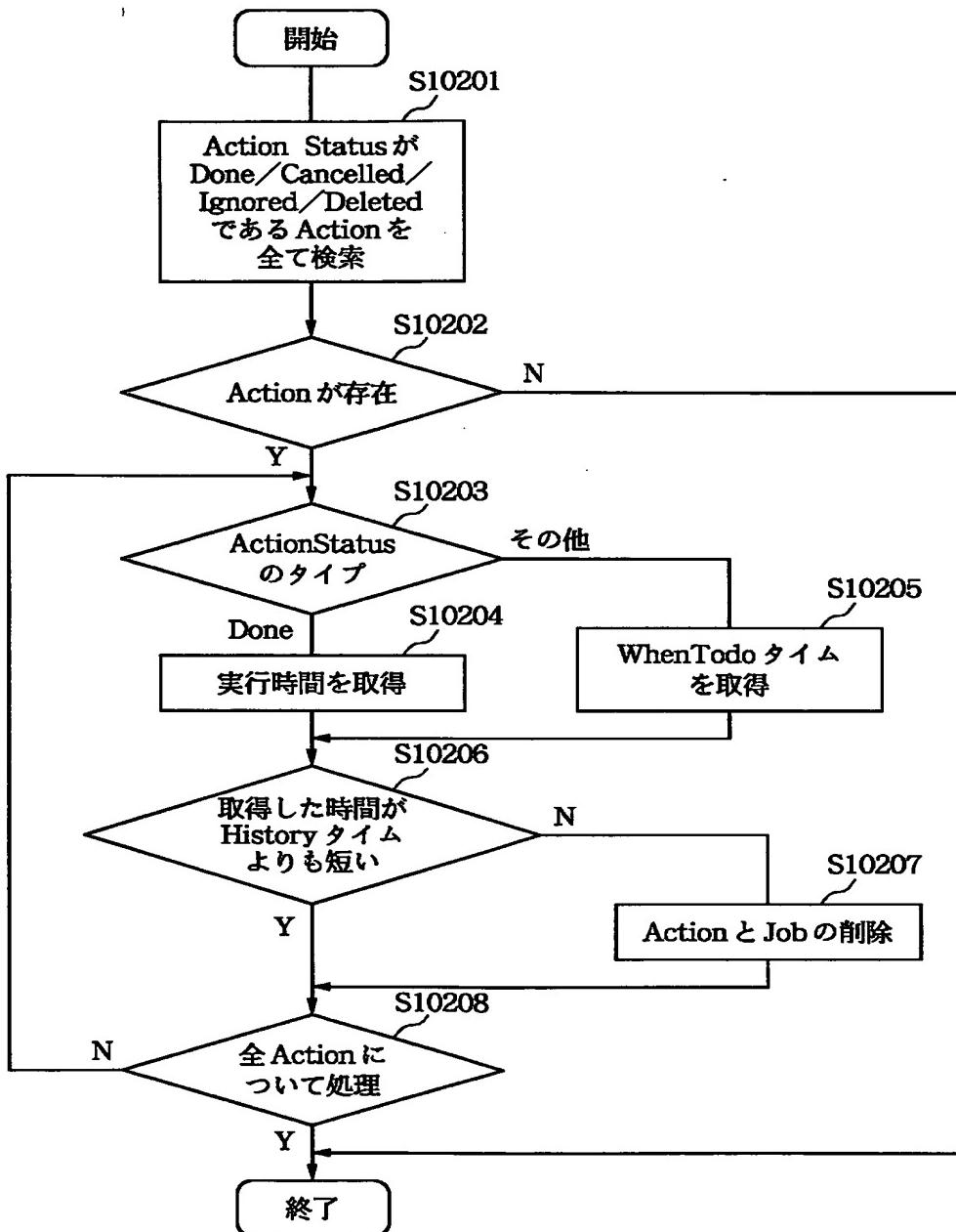
【図100】



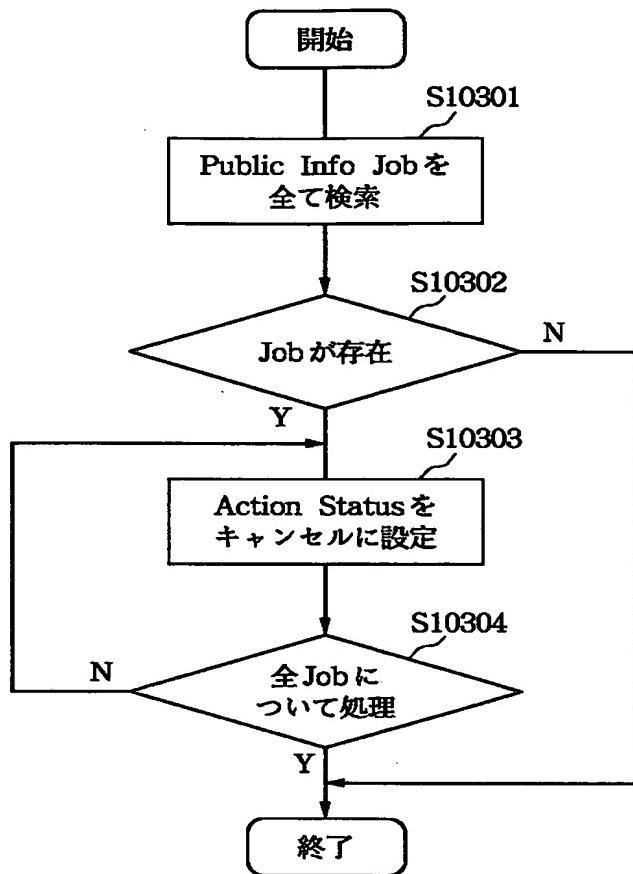
【図101】



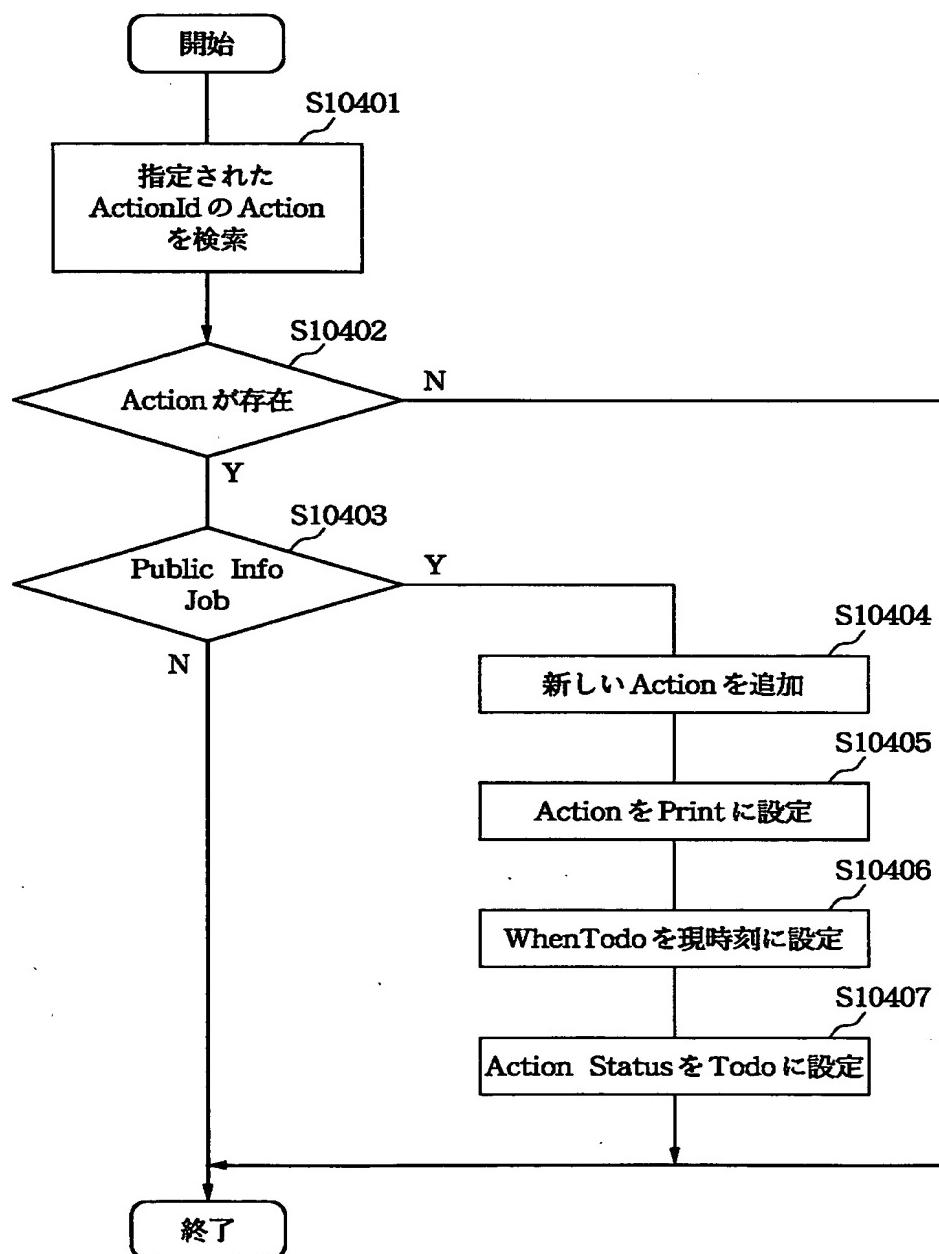
【図102】



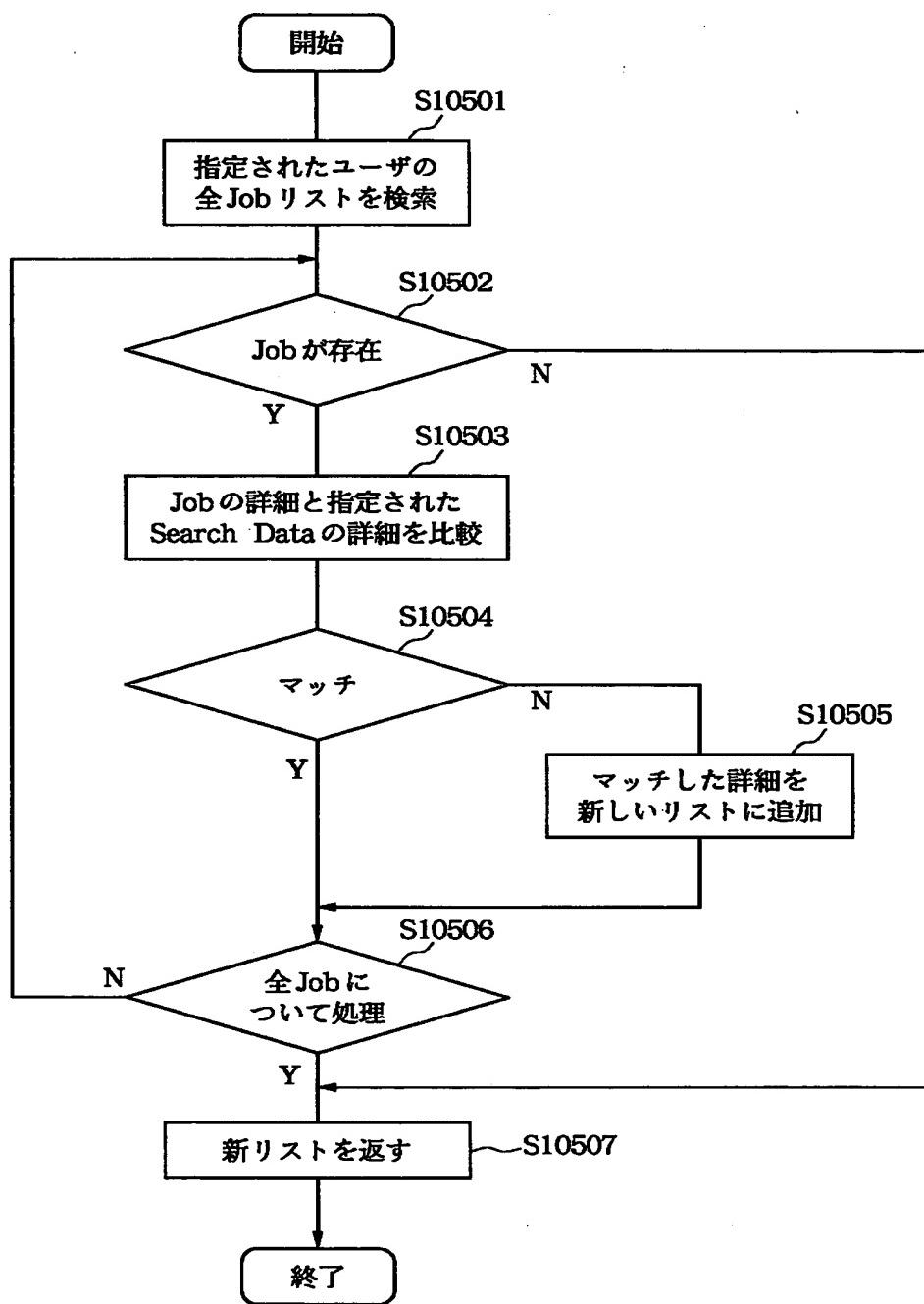
【図103】



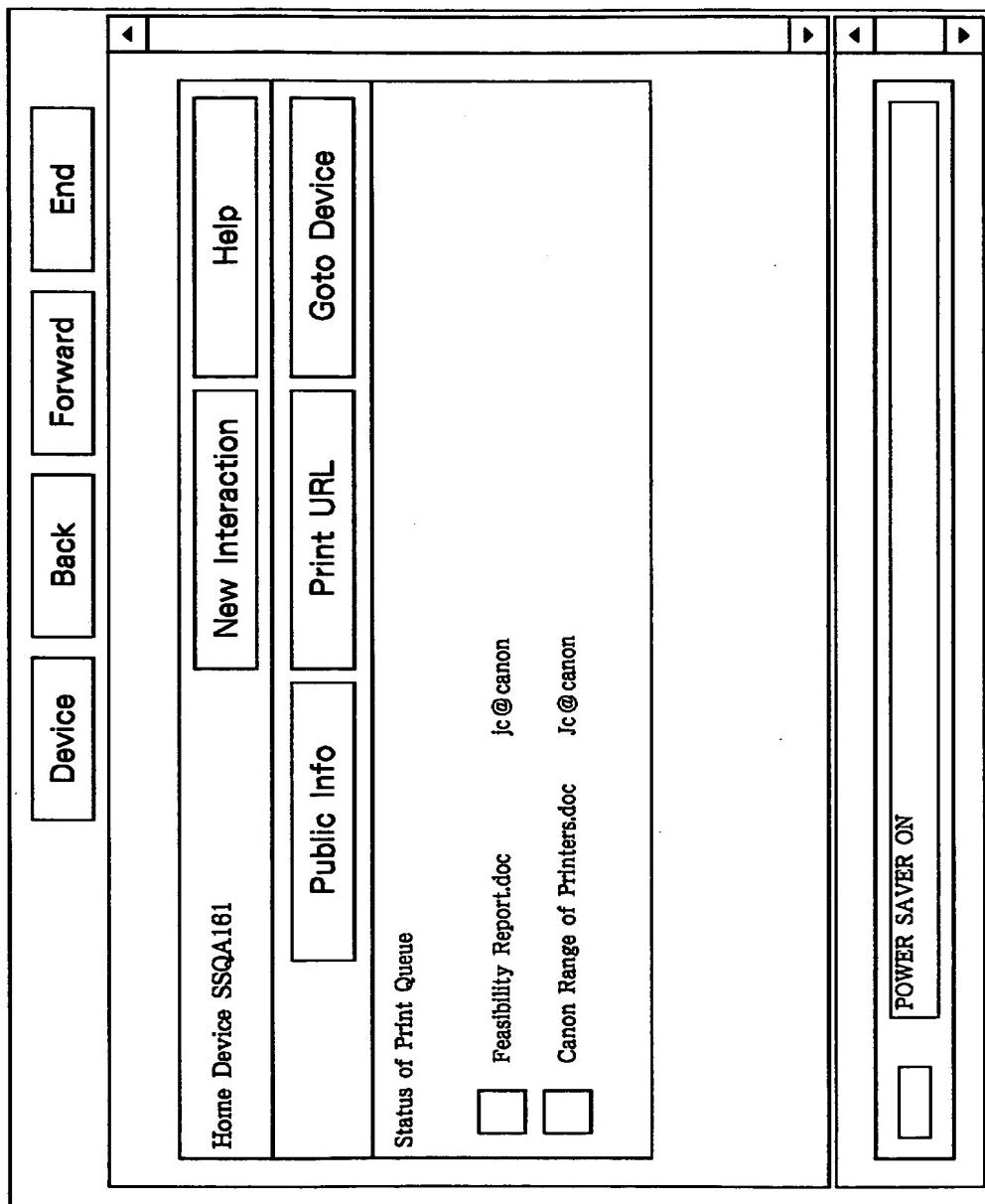
【図104】



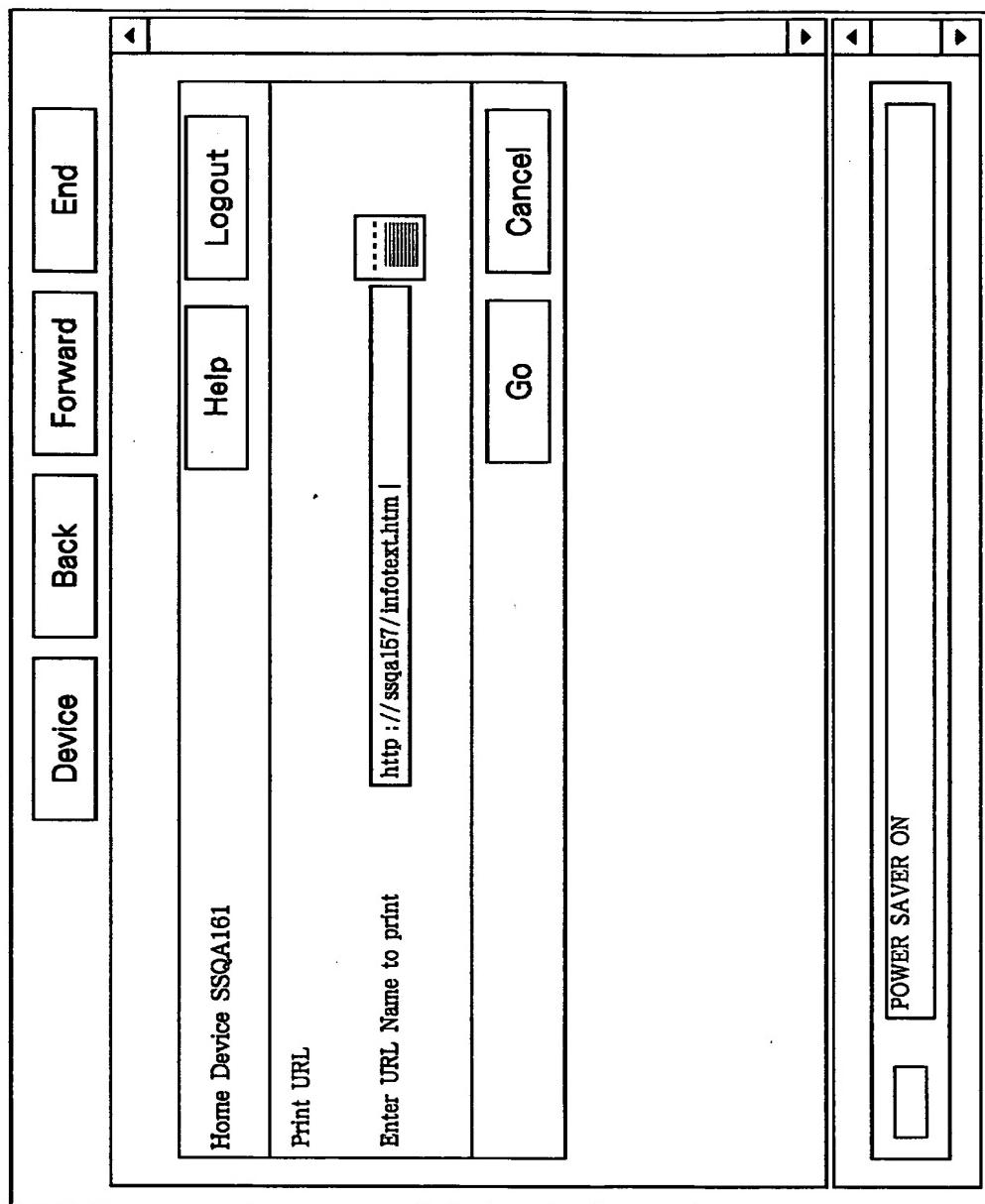
【図105】



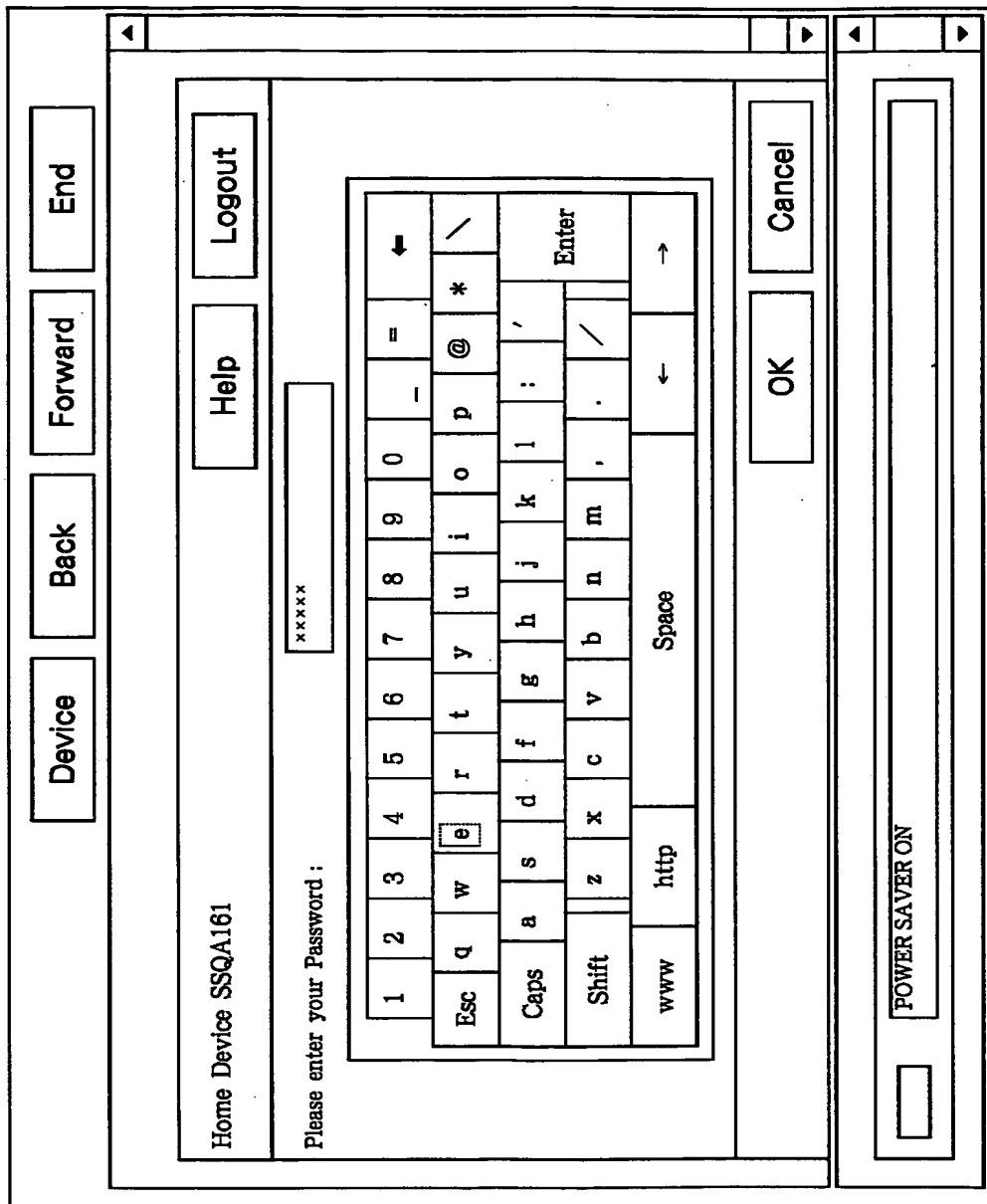
【図106】



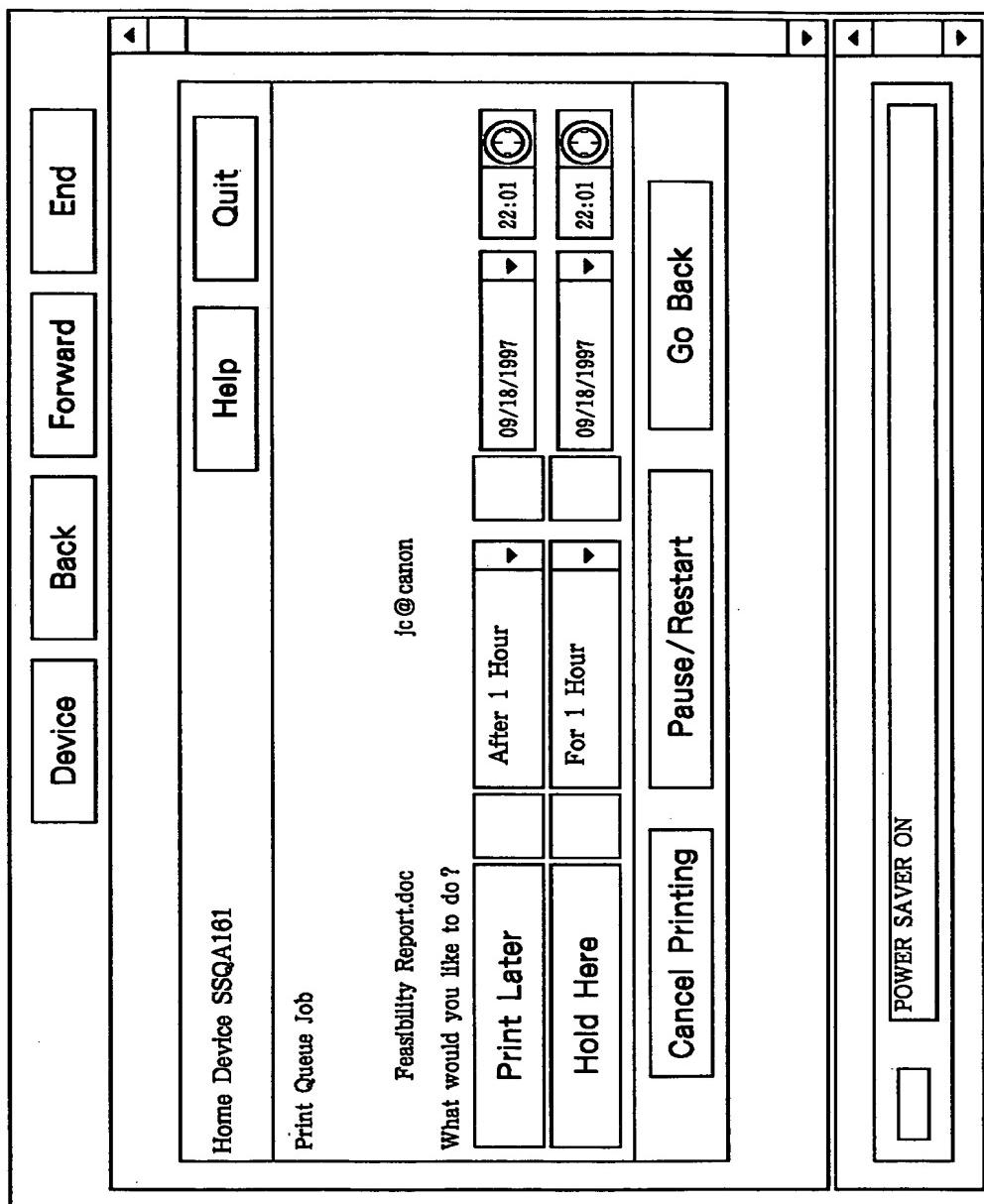
【図107】



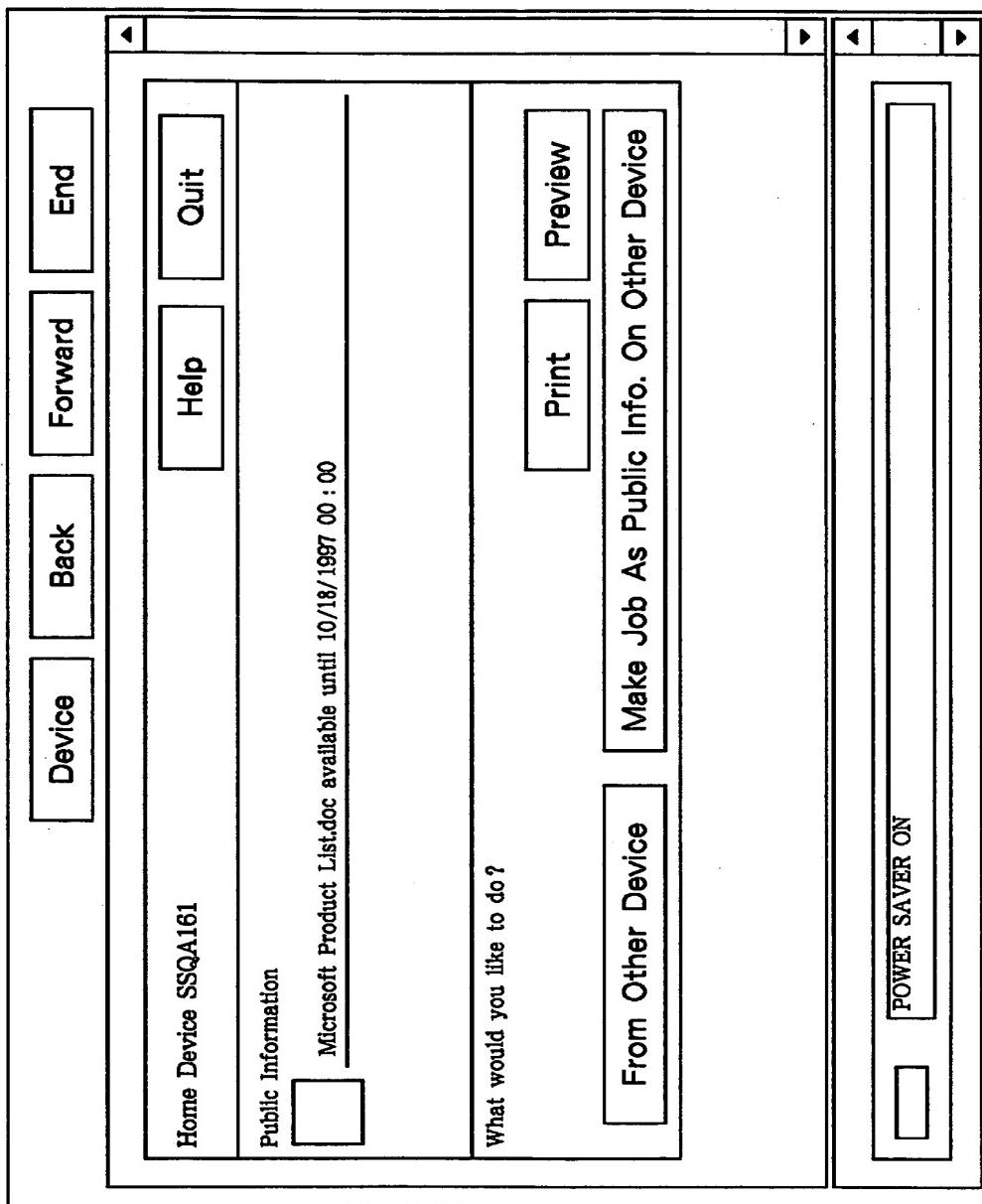
【図108】



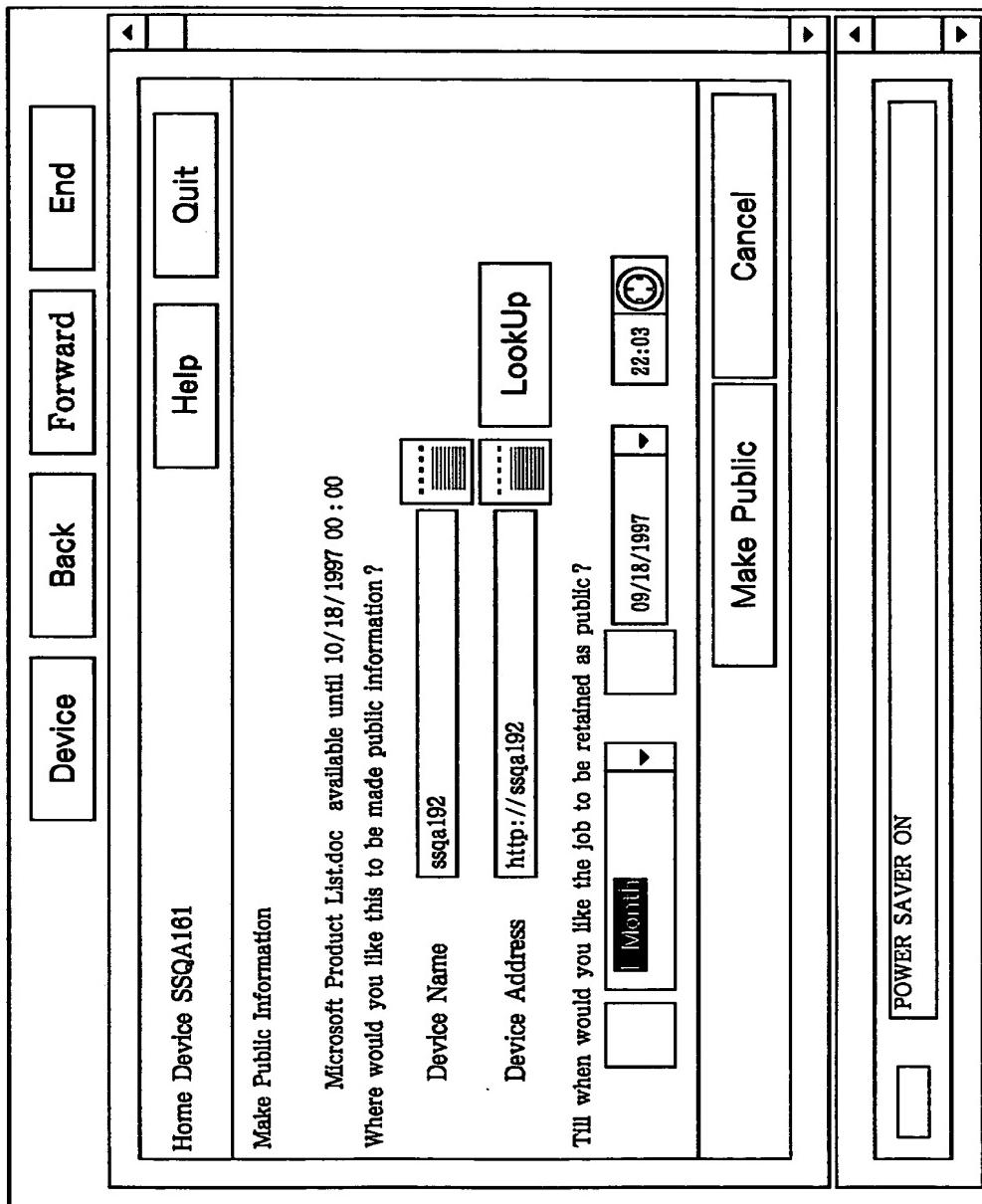
【図109】



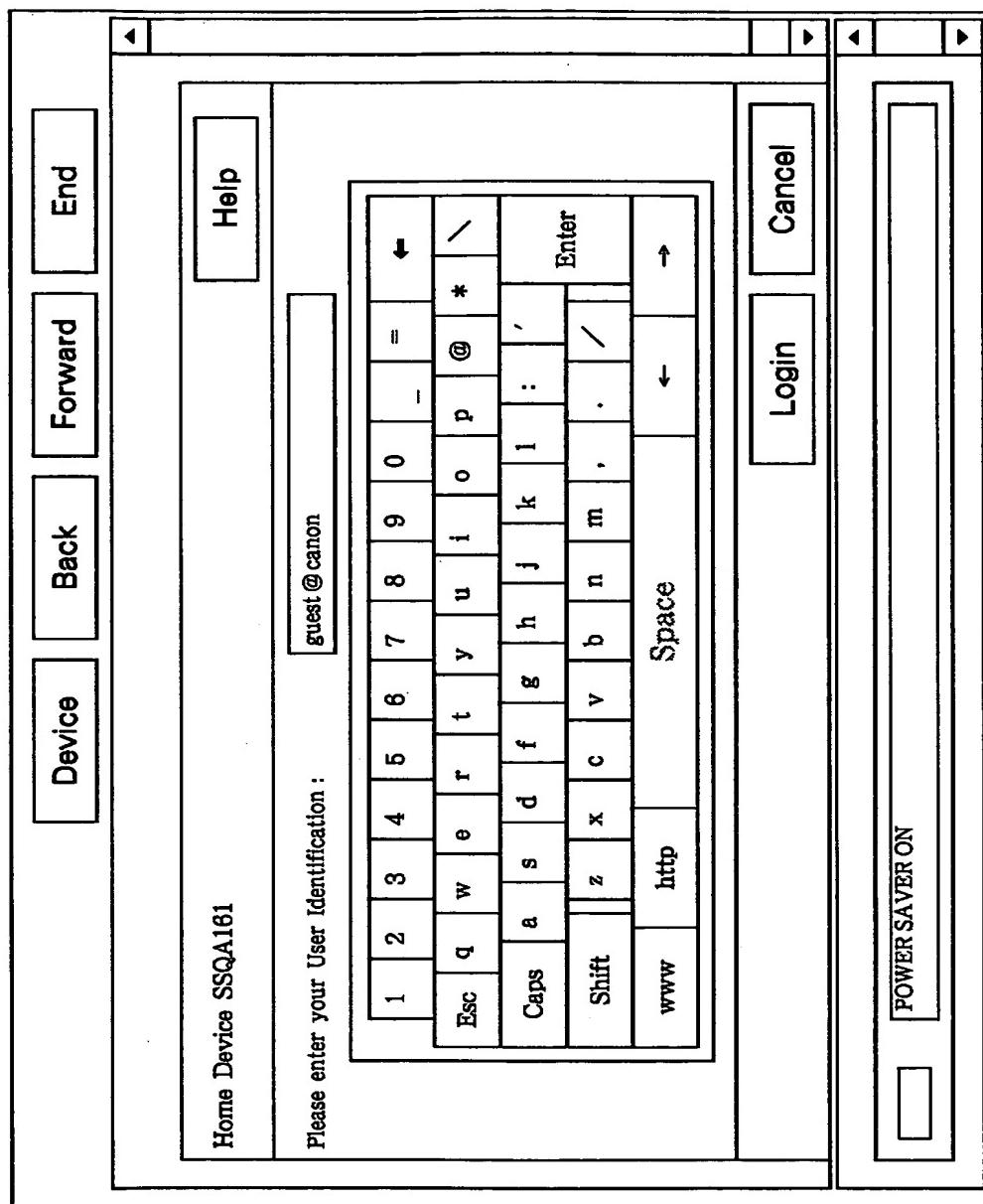
【図110】



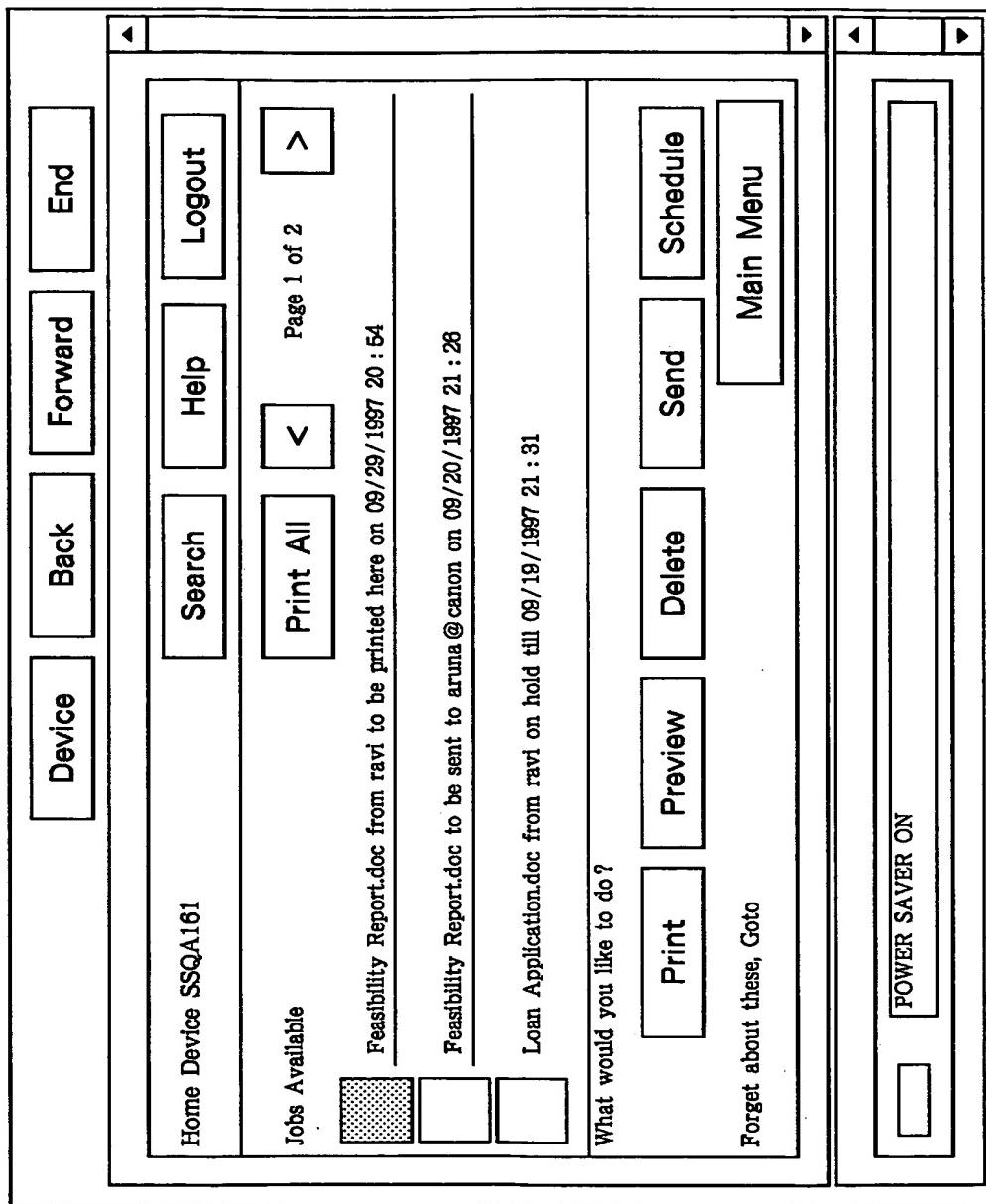
【図111】



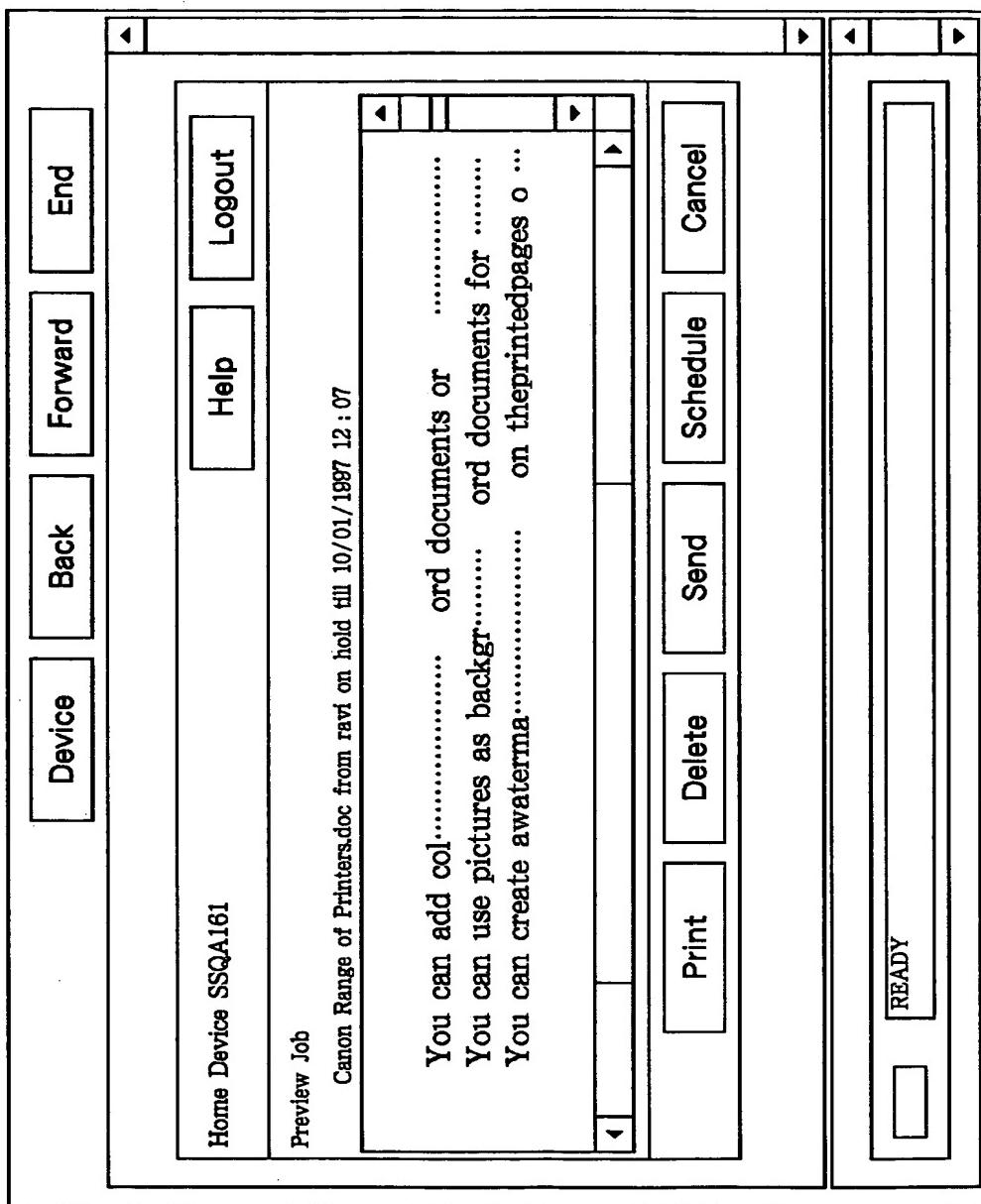
【図112】



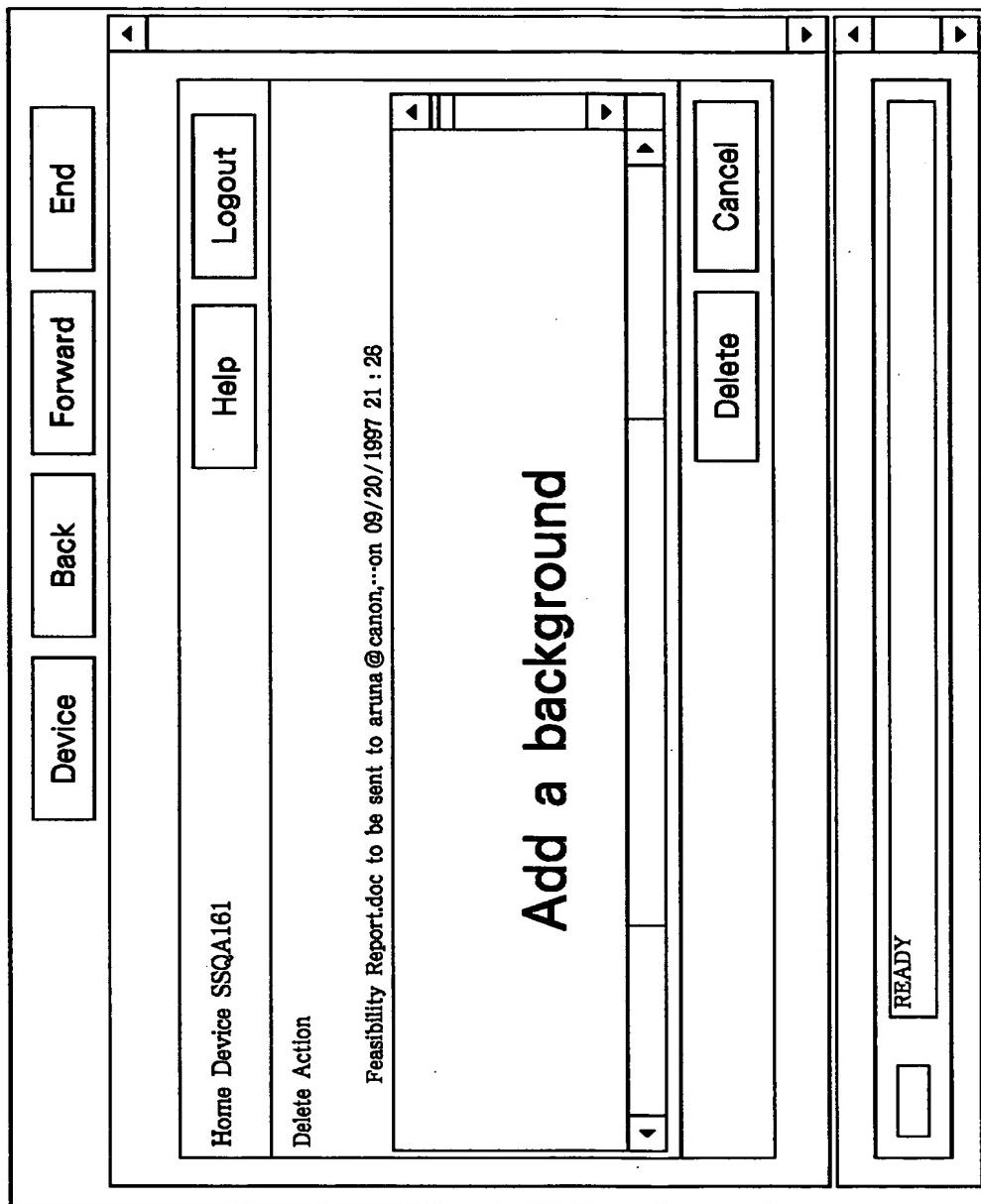
【図113】



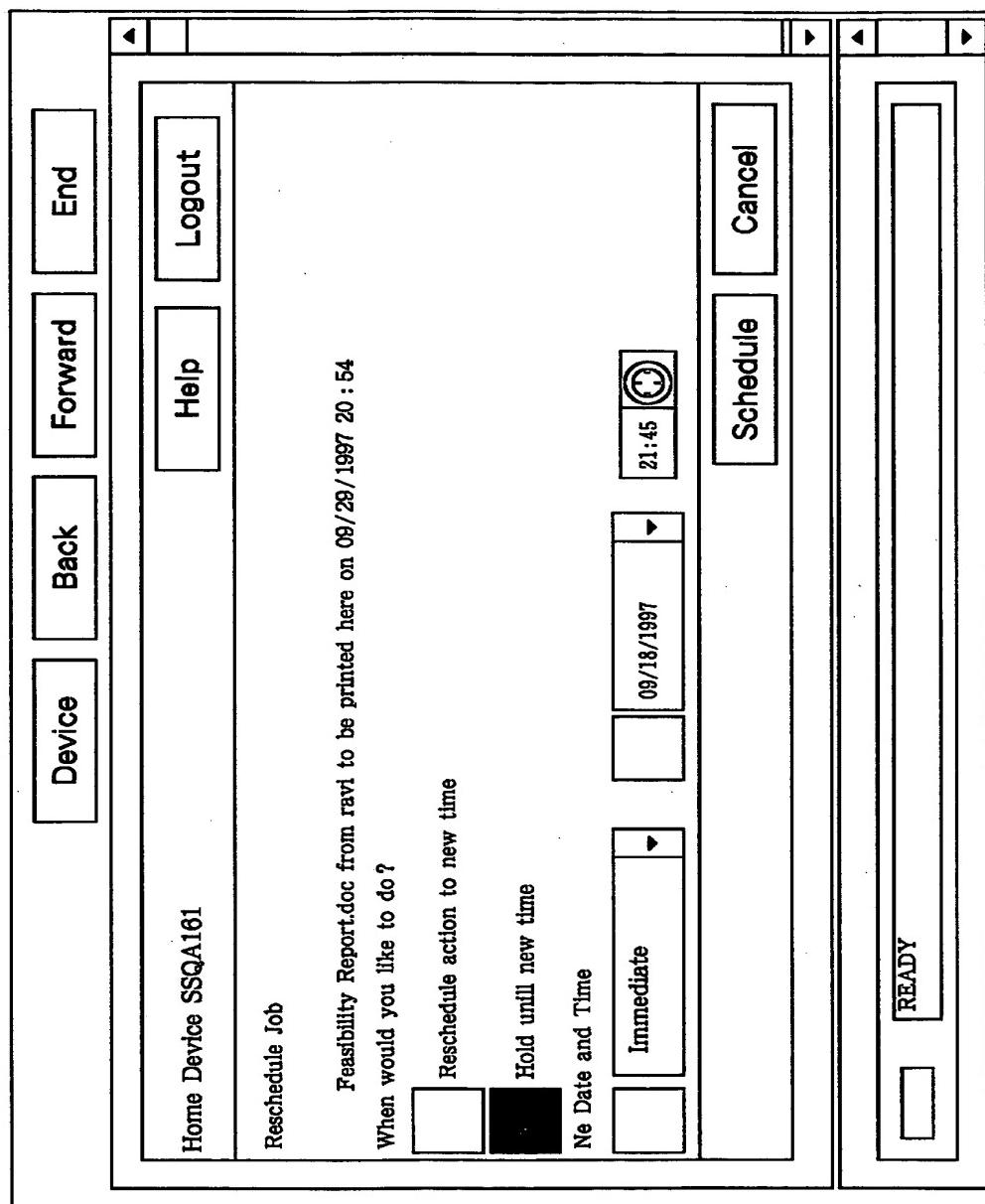
【図114】



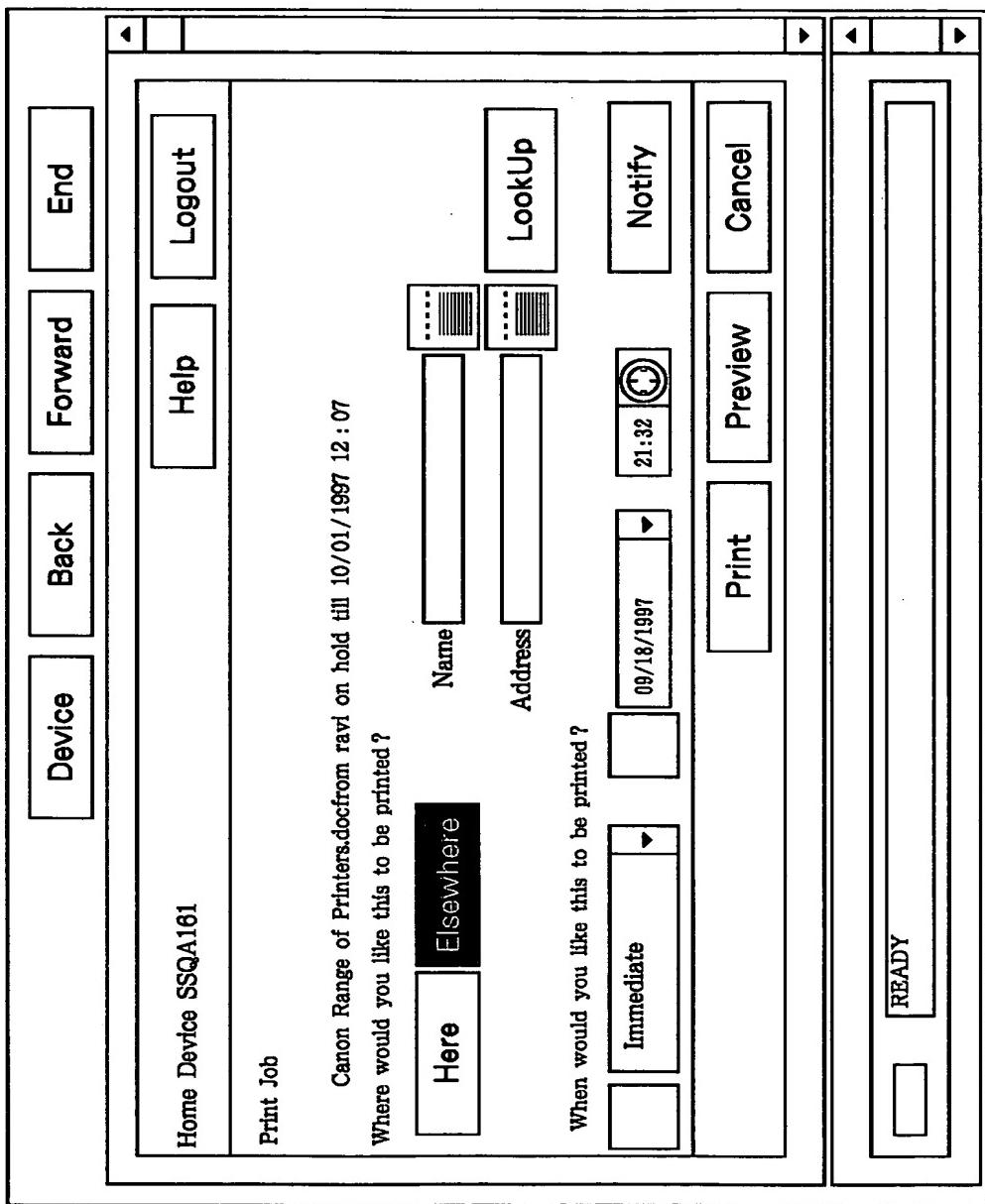
【図115】



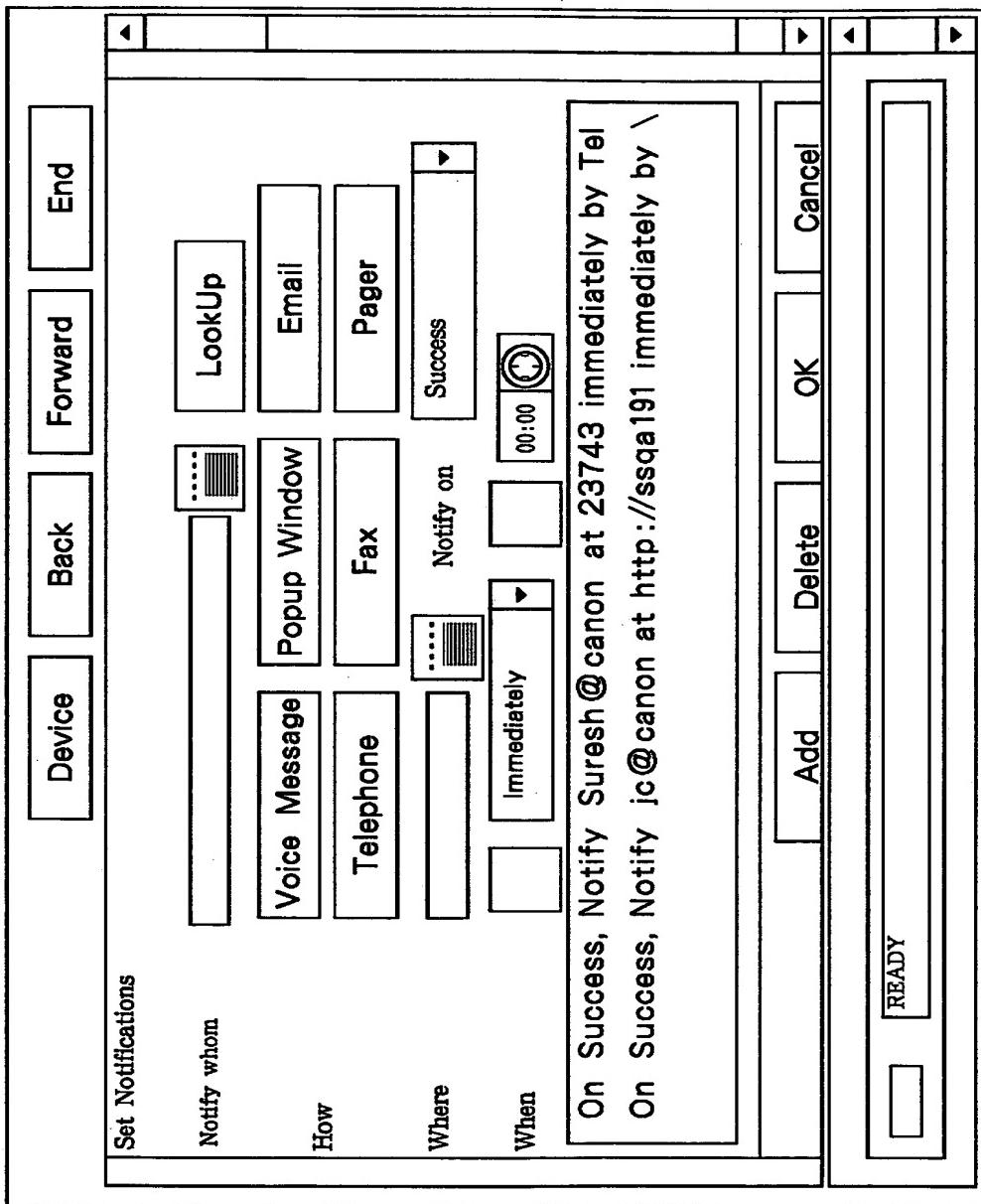
【図116】



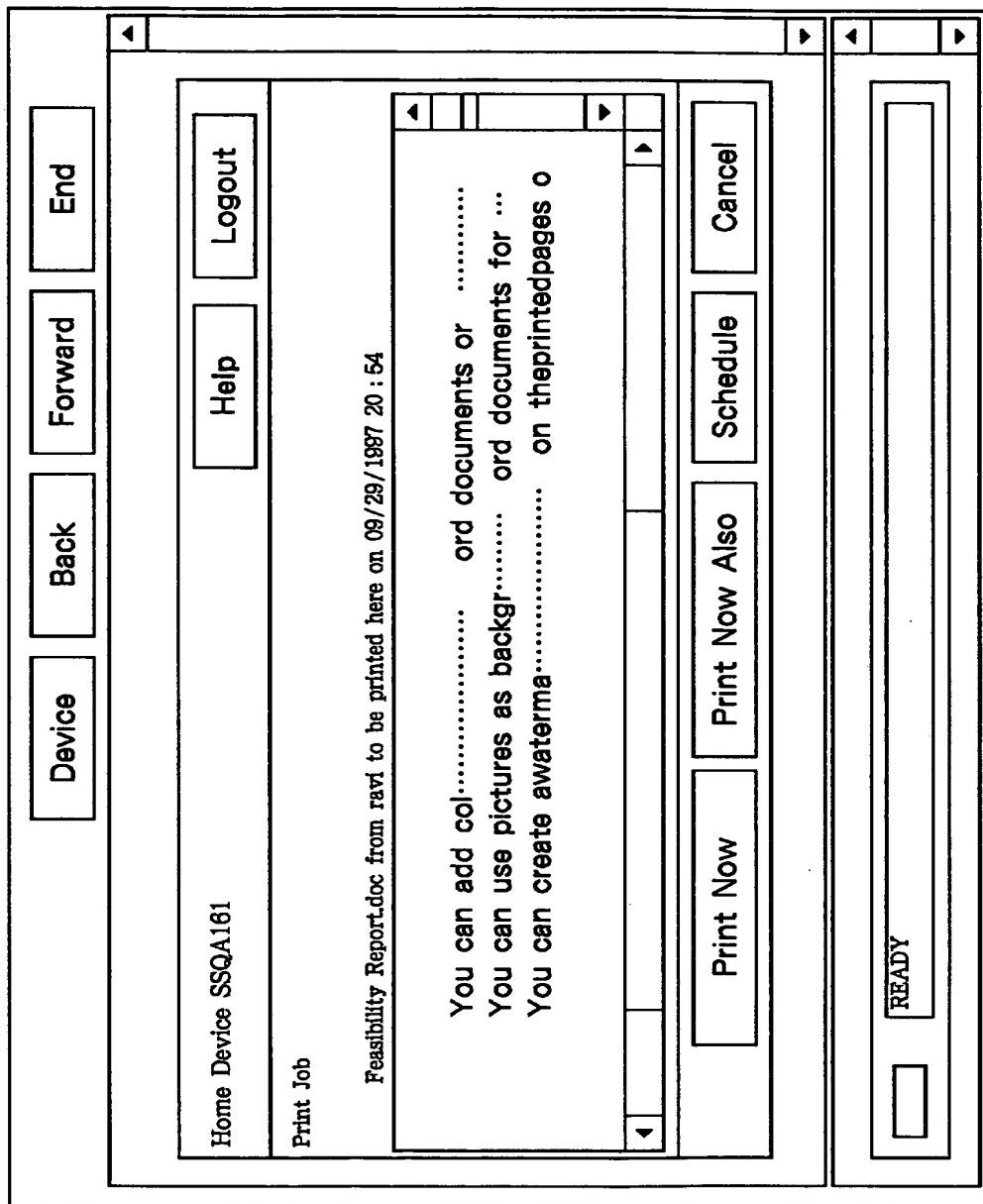
【図117】



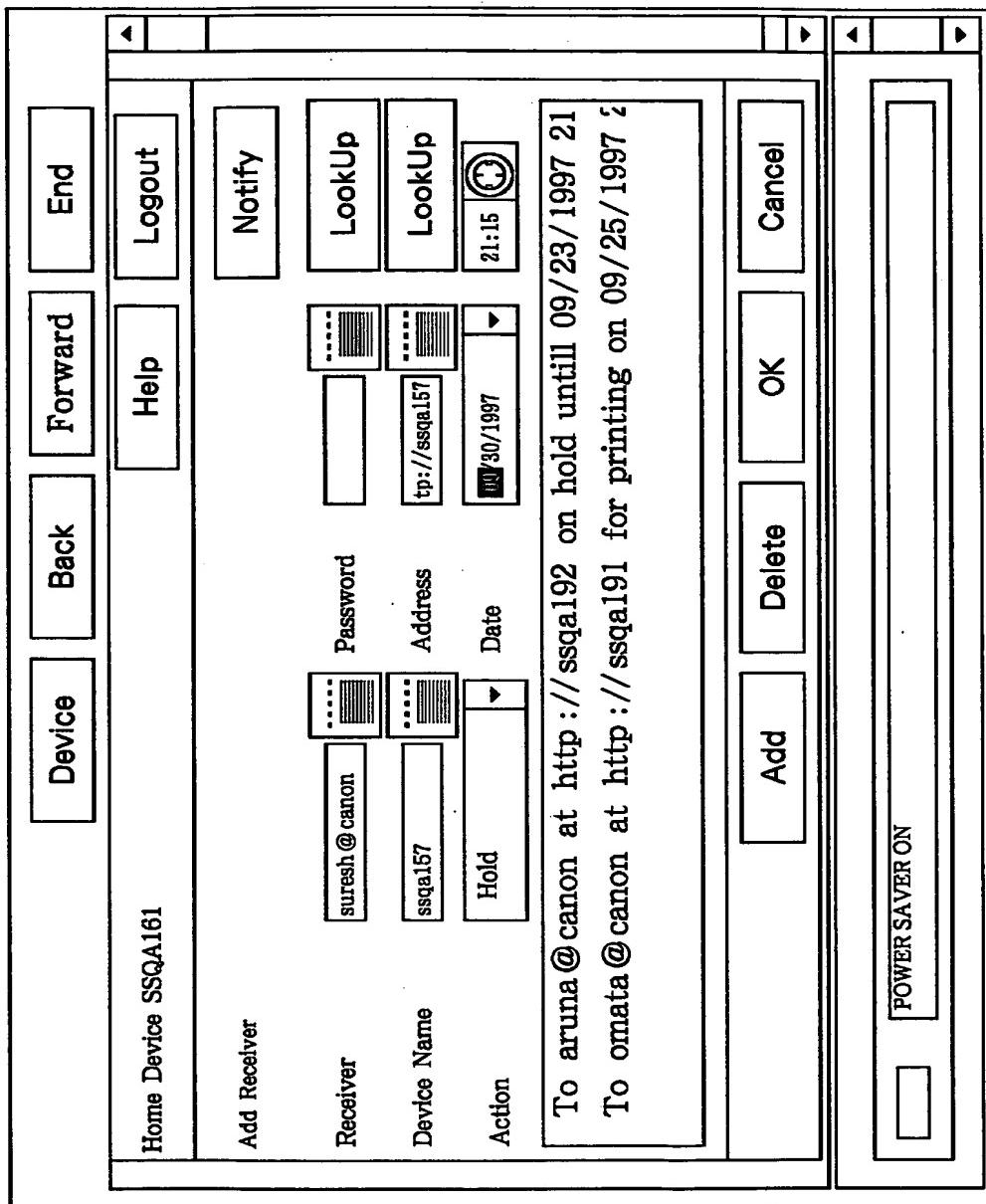
【図118】



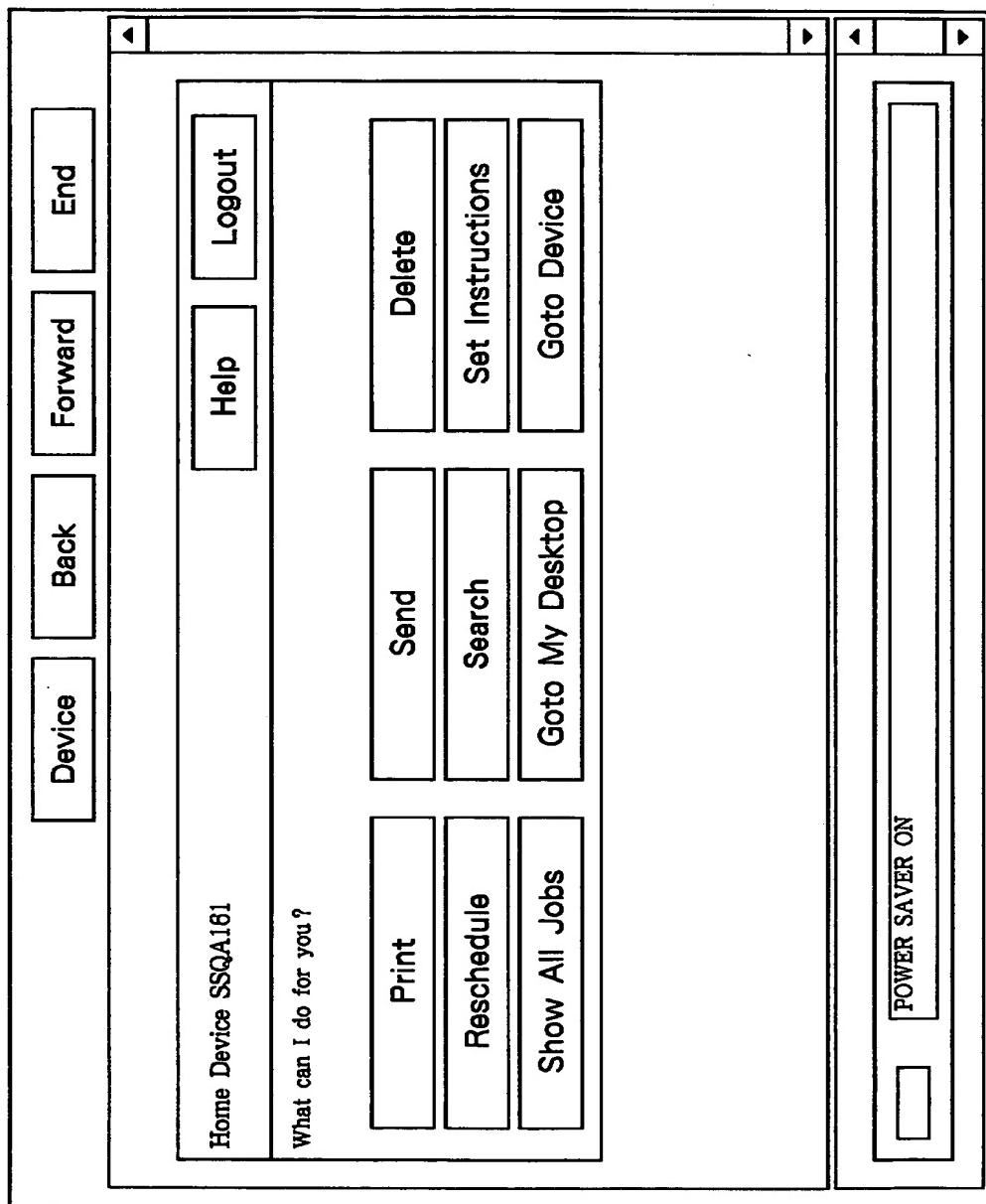
【図119】



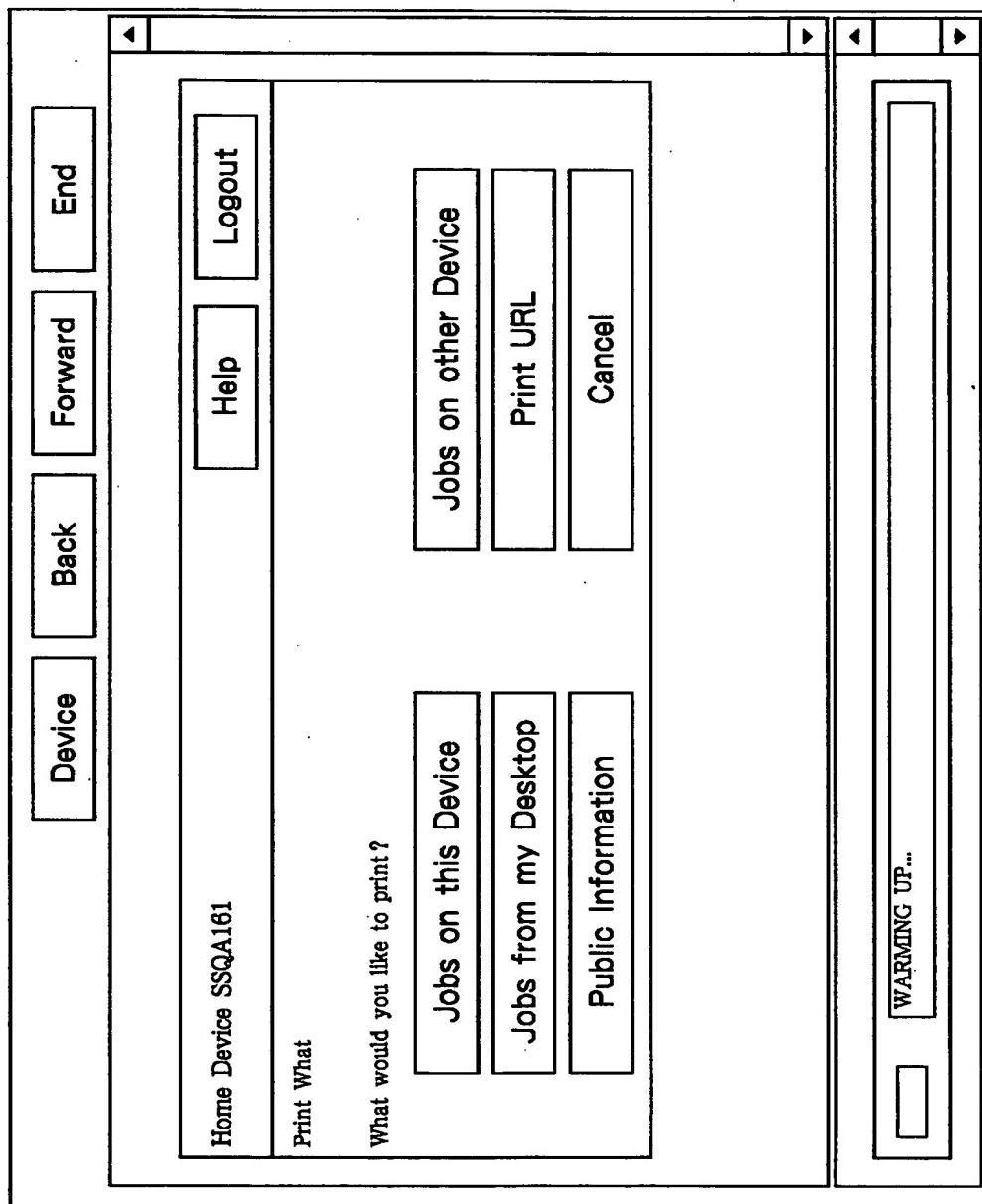
【図120】



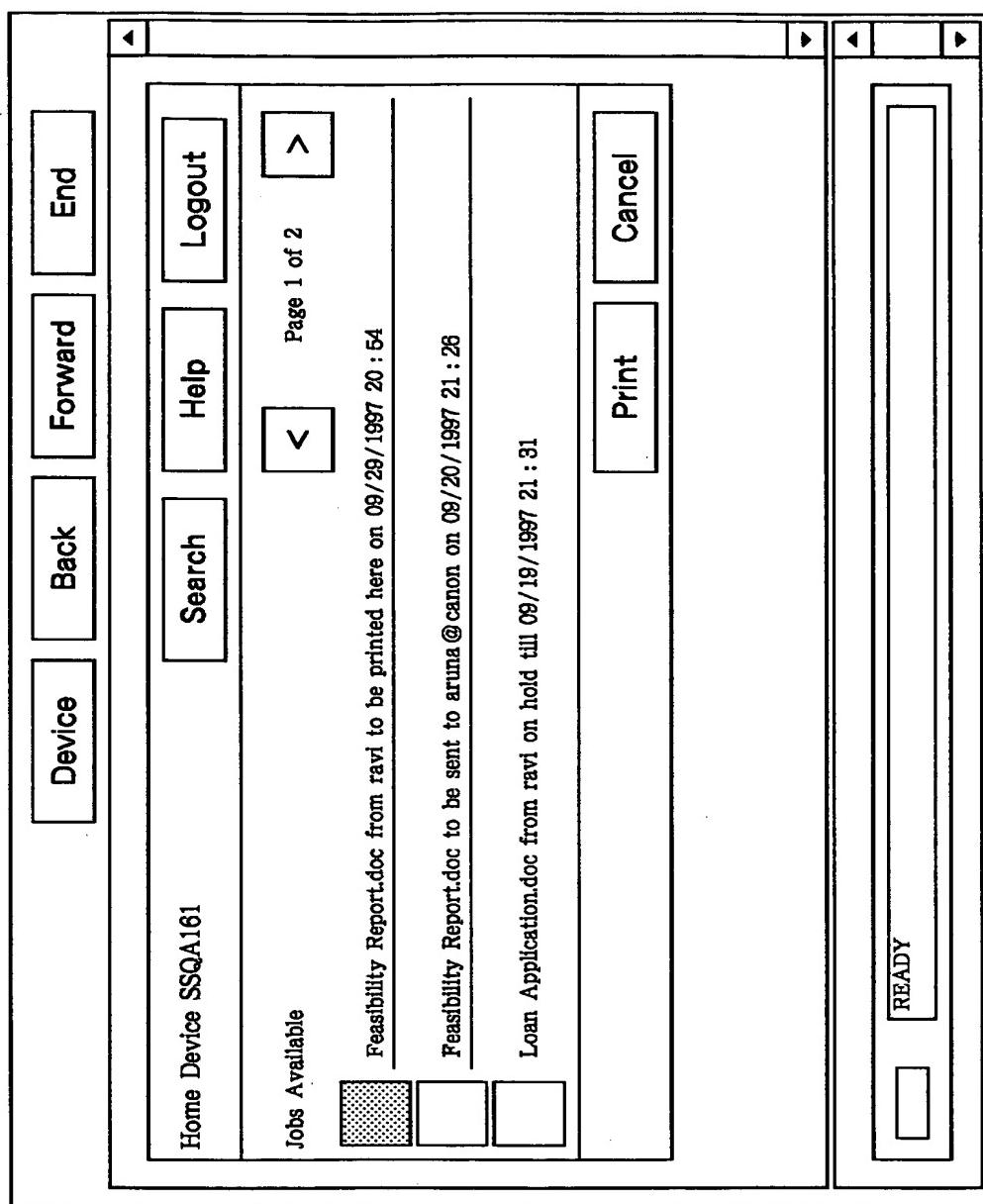
【図121】



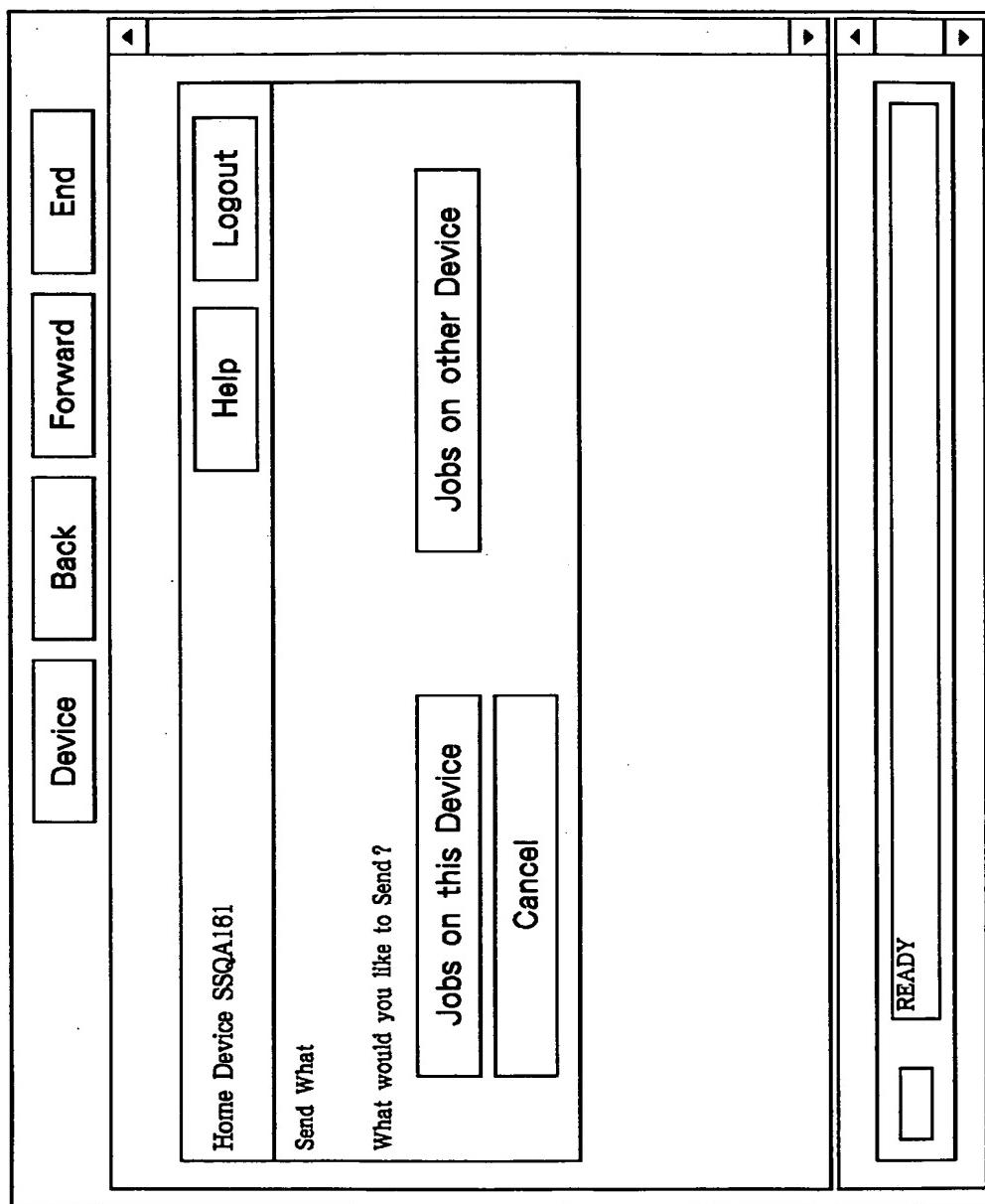
【図122】



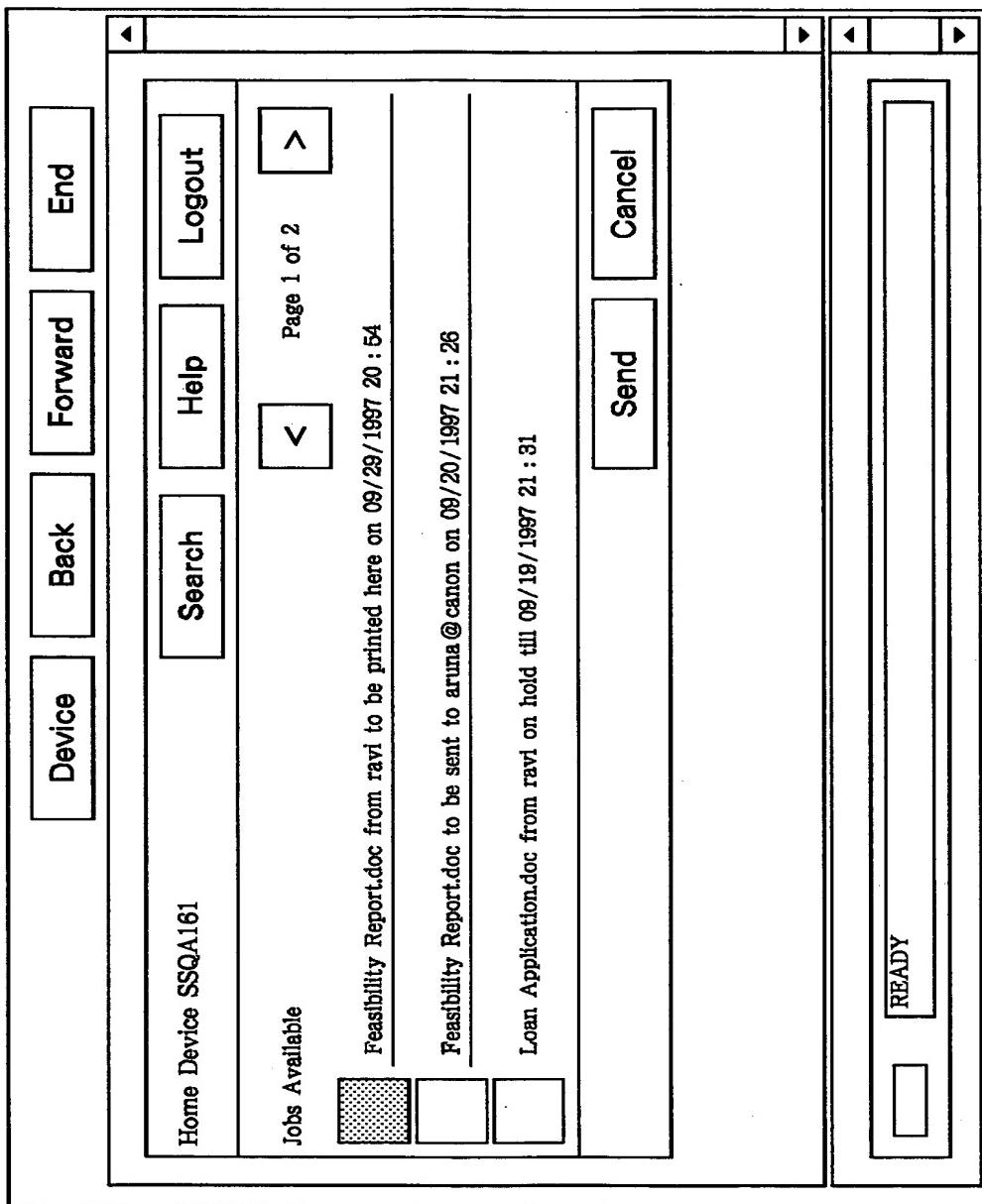
【図123】



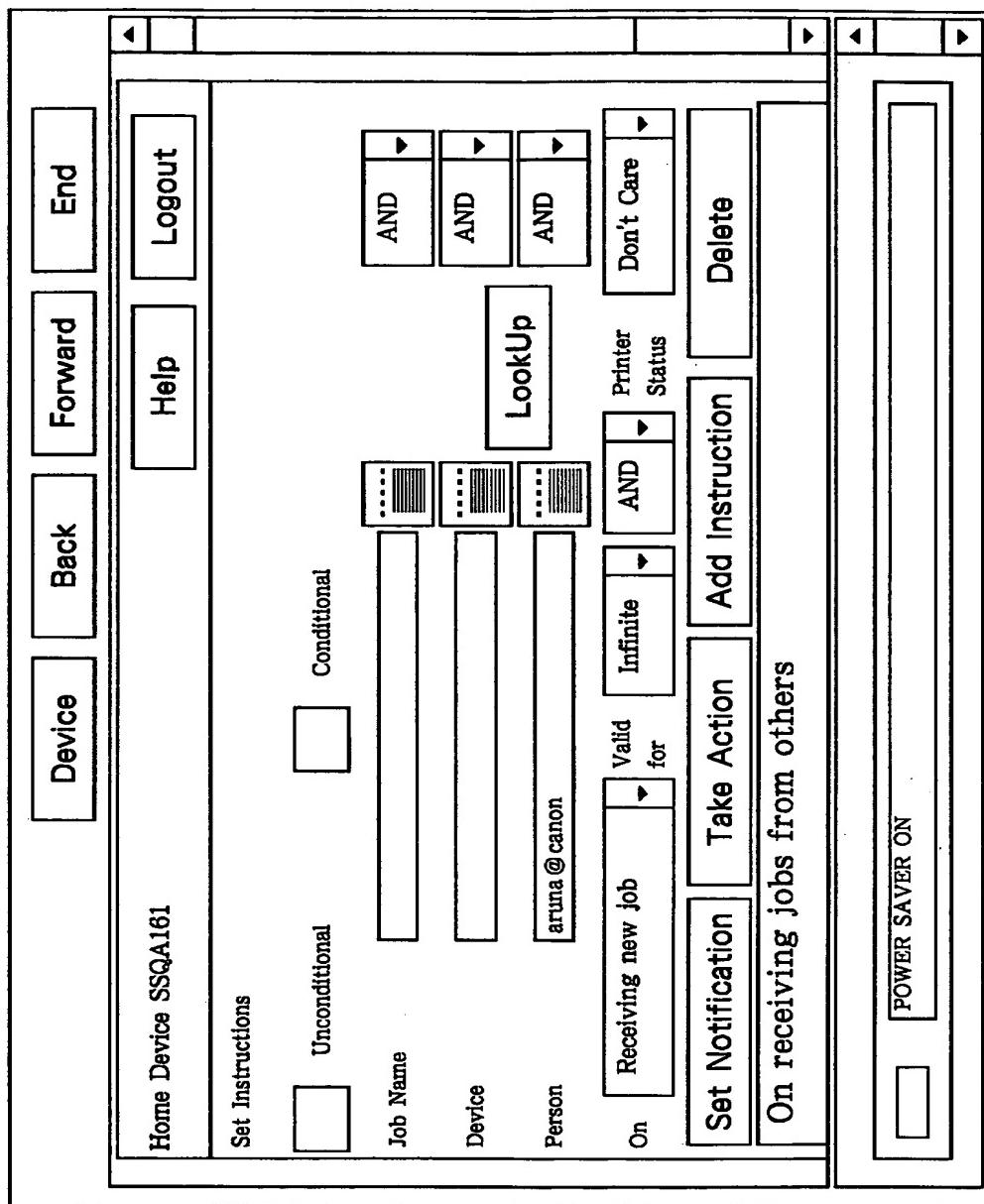
【図124】



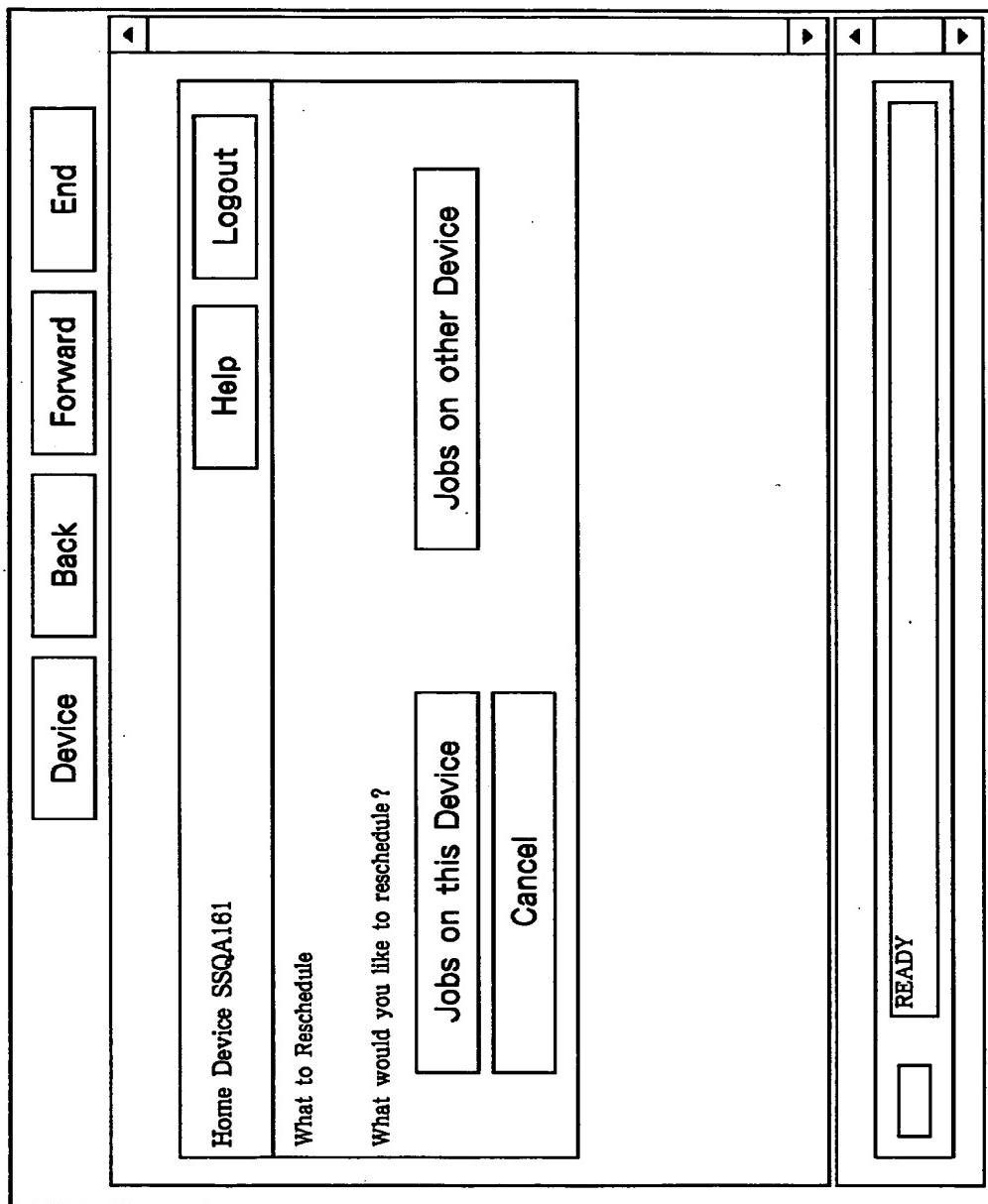
【図125】



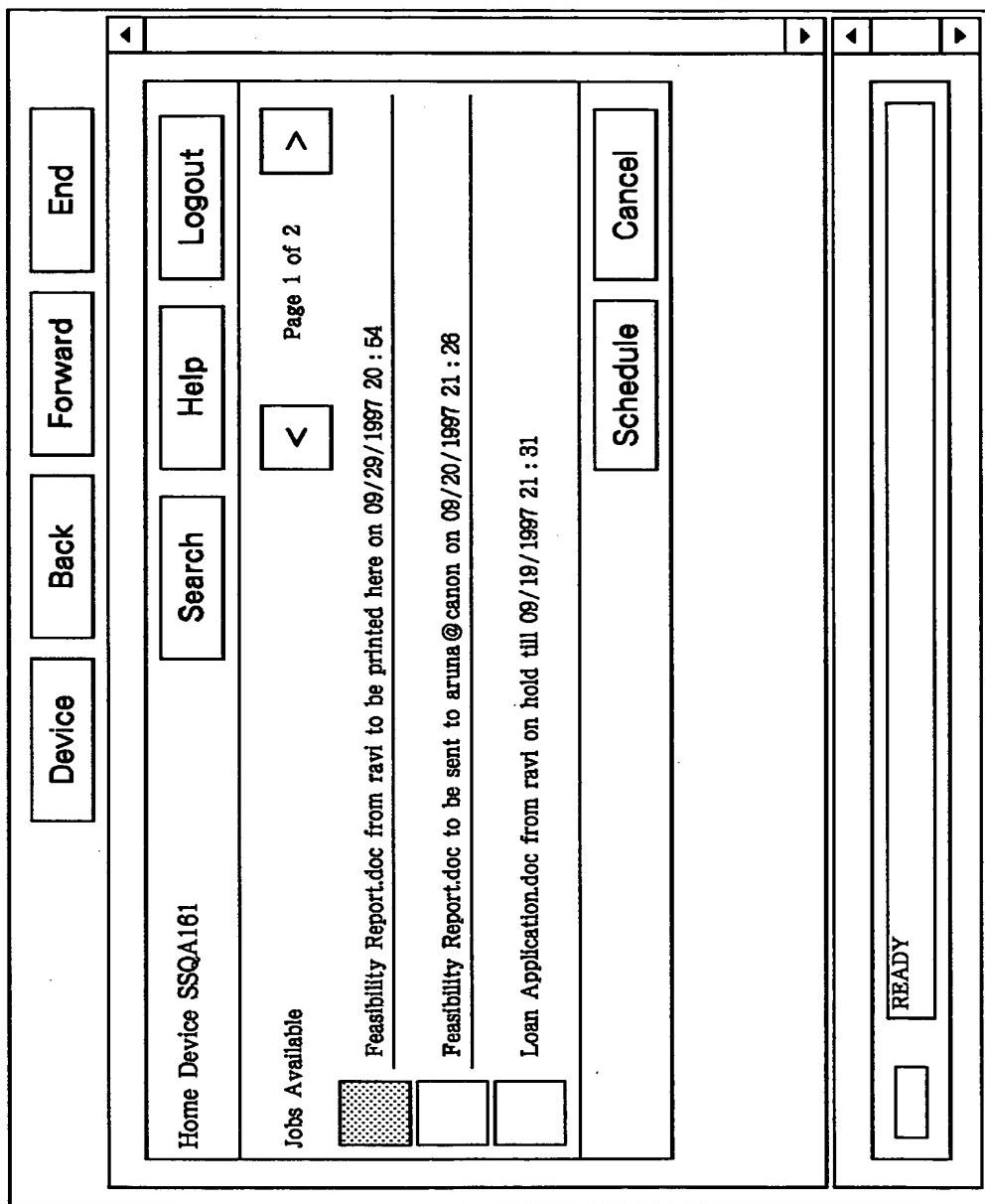
【図126】



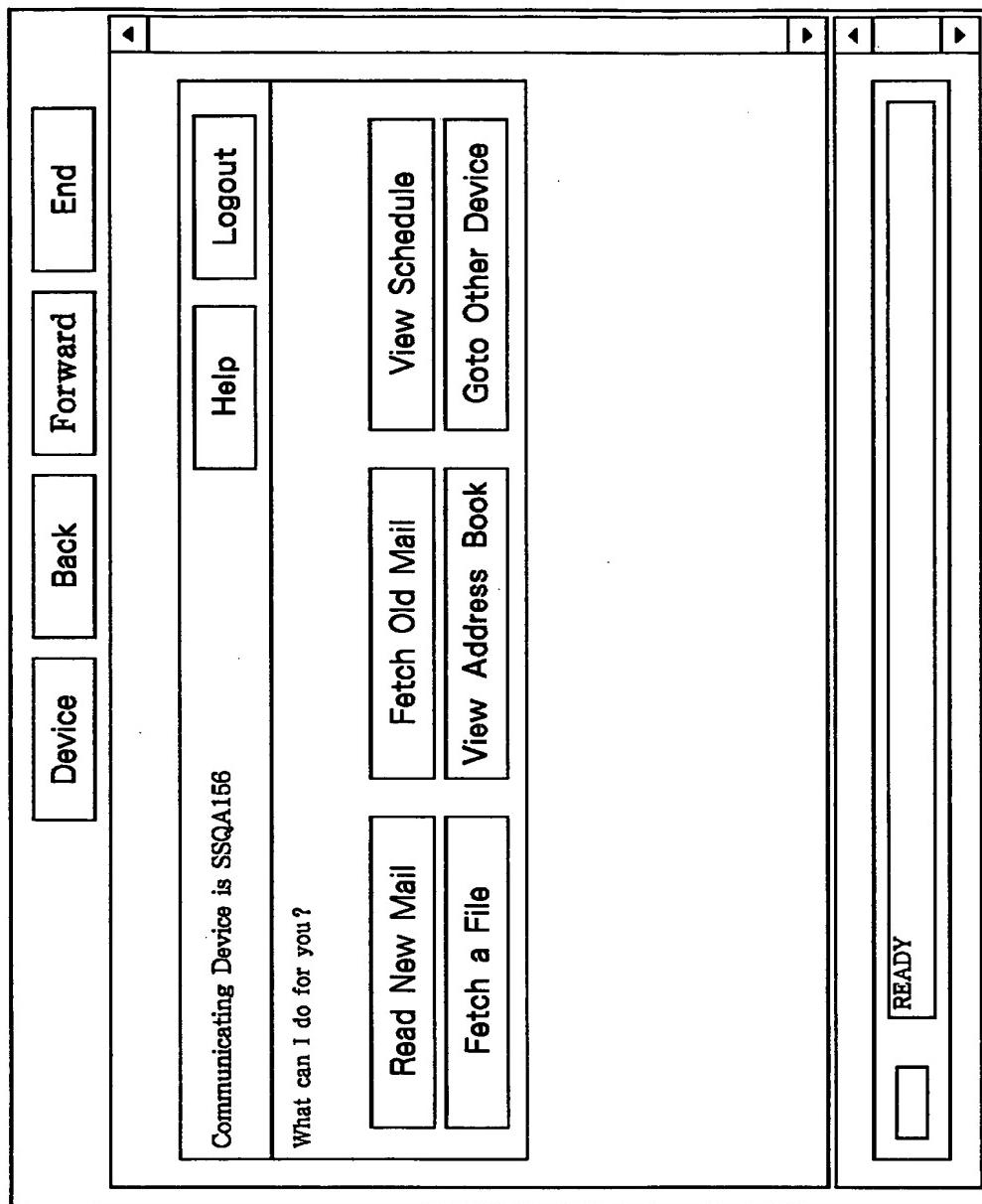
【図127】



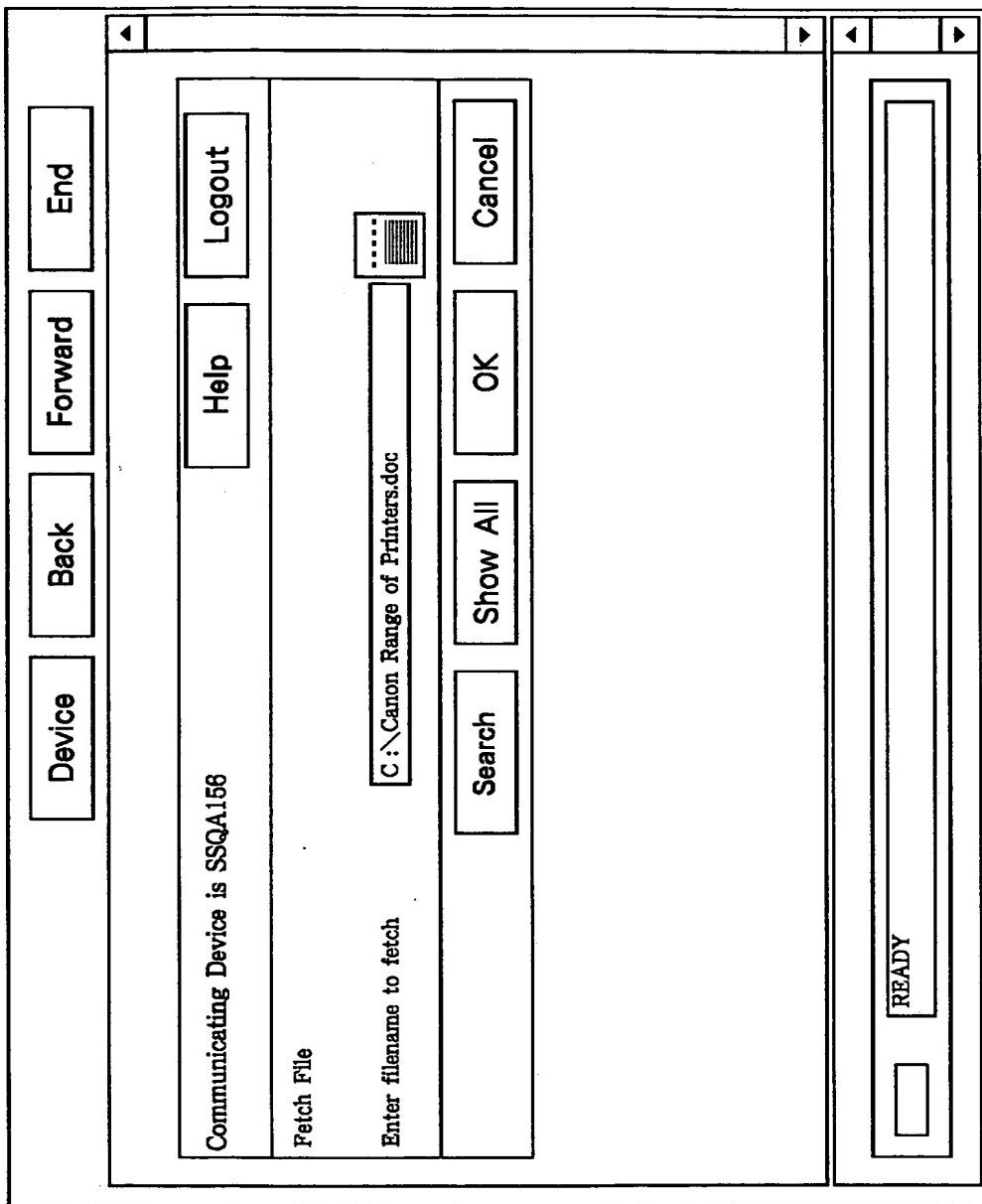
【図128】



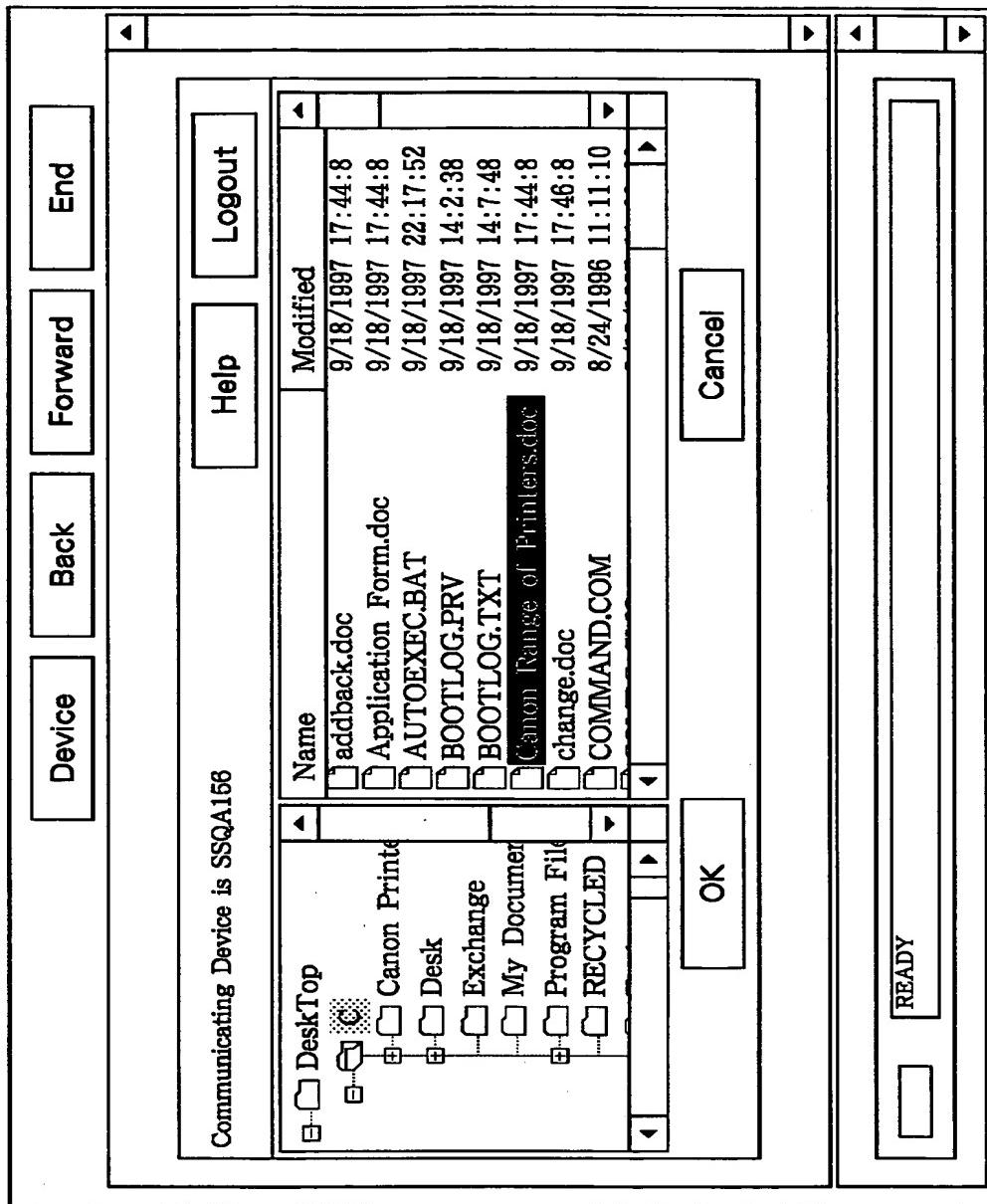
【図129】



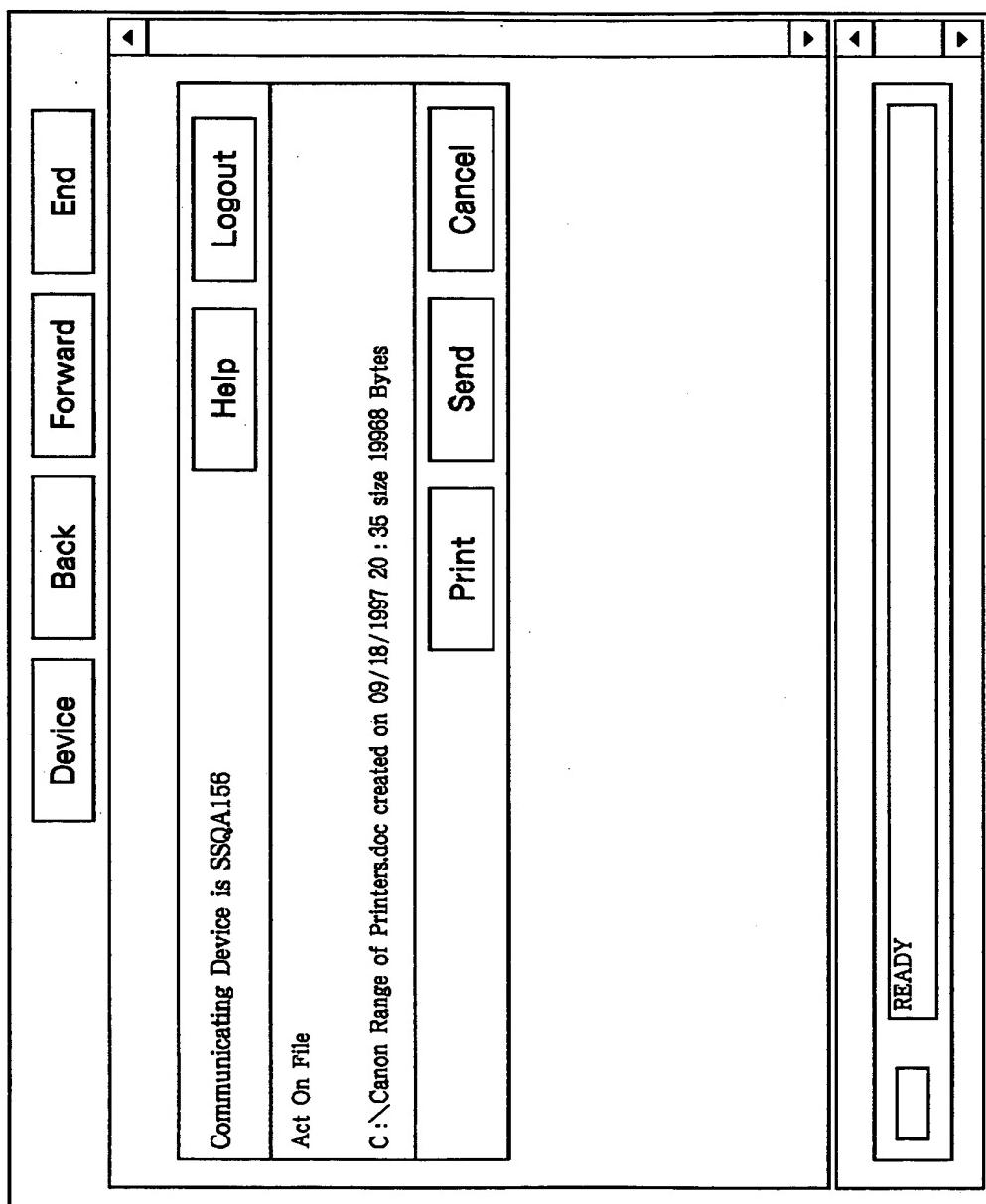
【図130】



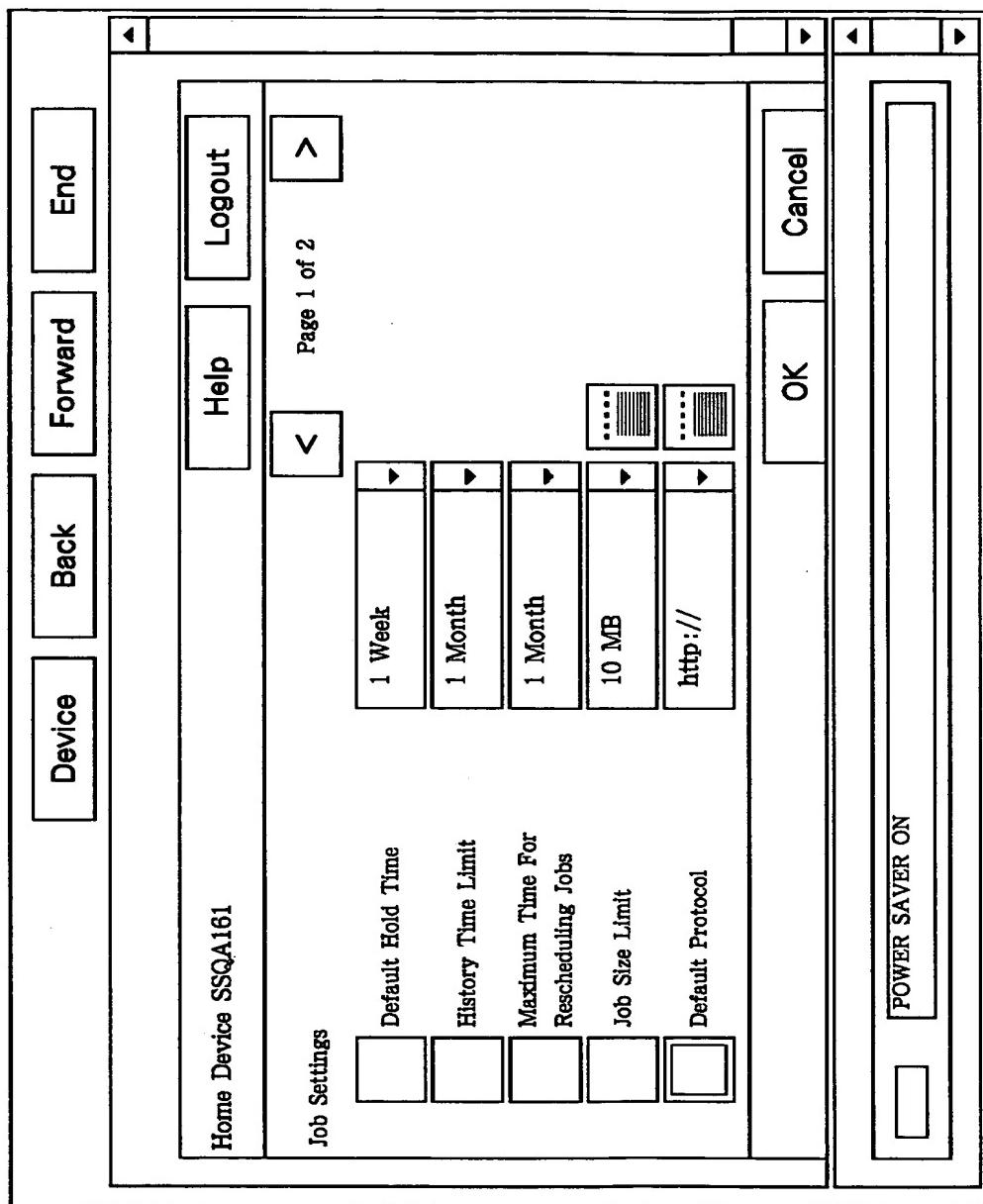
【図131】



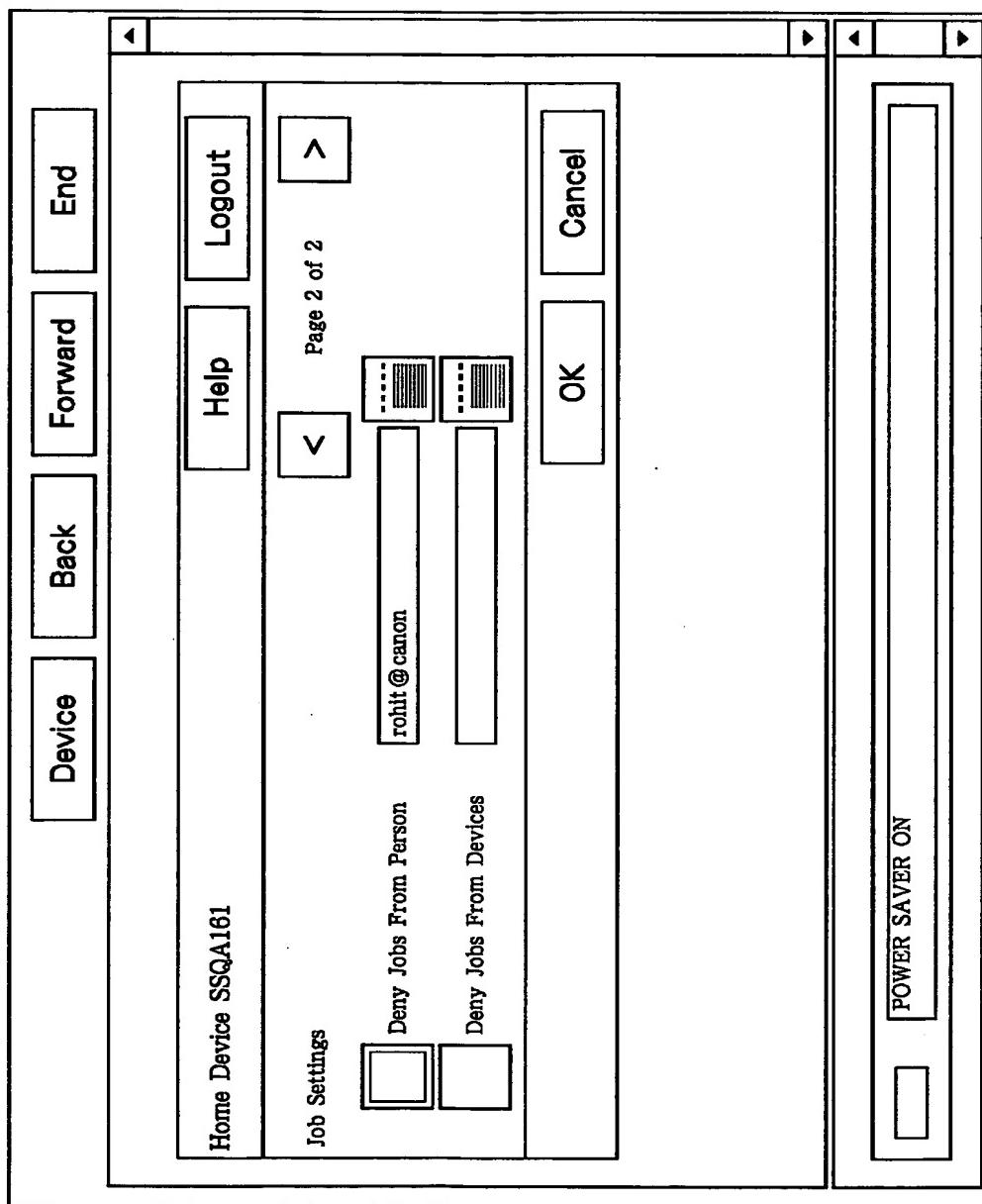
【図132】



【図133】



【図134】



【書類名】 要約書

【要約】

【課題】 処理対象情報に関する種々の条件が満たされたとき、所望の処理が行なえるようにする。

【解決手段】 情報処理方法に、処理対象情報の属性を指定する属性指定工程と、処理対象情報に対する処理を指定する処理指定工程と、実行すべき命令を設定する設定工程と、前記属性指定工程により指定された属性の処理対象情報に対して前記処理指定工程により指定された処理が実行された場合に、前記設定工程より設定された命令を実行する実行工程とを備える。

【選択図】 図77

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000001007

【住所又は居所】 東京都大田区下丸子3丁目30番2号

【氏名又は名称】 キヤノン株式会社

【代理人】 申請人

【識別番号】 100069877

【住所又は居所】 東京都大田区下丸子3-30-2 キヤノン株式会社内

【氏名又は名称】 丸島 儀一

出願人履歴情報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都大田区下丸子3丁目30番2号

氏 名 キヤノン株式会社